



**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**  
DEPARTAMENTO DE MECÁNICA DE MEDIOS  
CONTINUOS Y TEORÍA DE ESTRUCTURAS

**Desarrollo de un programa informático para el  
análisis dinámico de estructuras**

**PROYECTO FIN DE CARRERA**  
**INGENIERÍA TÉCNICA INDUSTRIAL MECÁNICA**

**Autor:** Alfonso Gago Rodríguez

**Tutor:** Carlos Santiuste Romero

Enero 2012



## ***Agradecimientos***

A mi tutor Carlos Santiuste, cuyas enseñanzas y guía siempre con la mejor de las actitudes y disponibilidad han hecho posible la realización de este proyecto.

A mis padres, hermana y abuela, por su apoyo incondicional ante cualquier situación.

A mis amigos y compañeros con los que he recorrido y me he apoyado en esta etapa de mi vida, tanto en los buenos como malos momentos.

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>1.1. Motivación</b>	<b>2</b>
<b>1.2. Objetivos</b>	<b>4</b>
<b>1.3. Resumen</b>	<b>5</b>
<b>2. Antecedentes</b>	<b>6</b>
<b>2.1. Teoría del cálculo matricial</b>	<b>7</b>
2.1.1. Método de la rigidez	7
2.1.1.1. Introducción	7
2.1.1.2. Procedimiento para el cálculo matricial de estructuras	8
2.1.1.2.1. Identificación estructura	9
2.1.1.2.2. Matriz de rigidez y vector de cargas equivalentes	10
2.1.1.2.3. Cargas nodales	14
2.1.1.2.4. Rotación de ejes en el plano	15
2.1.1.2.5. Matriz de rigidez global de la estructura	16
2.1.1.2.6. Matriz de cargas global de la estructura	18
2.1.1.2.7. Condiciones de contorno o borde y cálculo de reacciones en la estructura	19
<b>2.2. MatLab</b>	<b>20</b>
2.2.1. Introducción	20
2.2.2. Entorno de trabajo MatLab	20
2.2.2.1. Escritorio de MatLab	21
2.2.2.2. Command Window	22
2.2.2.3. Command History Browser	22
2.2.2.4. Current Directory Browser o Current Folder	22
2.2.2.5. Workspace y Array Editor o Variable Editor	23
2.2.2.6. Editor/Debugger	23
2.2.3. Interfaz gráfica GUIDE	24
2.2.3.1. Estructura de los gráficos de MatLab	25
2.2.3.1.1. Objetos gráficos de MatLab	25

2.2.3.1.2. <i>Identificadores (Handles)</i>	25
2.2.3.2. Propiedades de los objetos	25
2.2.3.2.1. <i>Funciones set y get</i>	26
2.2.3.3. Creación de controles gráficos: comando <i>uicontrol</i>	26
2.2.3.4. Tipos de <i>uicontrol</i>	27
2.2.3.5. Construcción de una interfaz gráfica	29
<b>2.3. Programas de cálculo de estructuras simplificado: Ed-Tridim</b>	30
2.3.1. Introducción	30
2.3.2. Planteamiento práctico	31
2.3.3. Resultados	33
2.3.3.1. Módulo Ejemplos	34
2.3.3.2. Módulo Ejercicios	35
<b>2.4. CypeCAD</b>	36
2.4.1. Introducción	36
2.4.2. Características diferenciales de CypeCAD	36
<b>3. Descripción del programa</b>	38
<b>3.1. Introducción</b>	39
<b>3.2. Organigrama</b>	40
<b>3.3. Procedimiento de cálculo</b>	42
3.3.1. Introducción de datos por el usuario	42
3.3.2. Cálculos previos	42
3.3.3. Cálculo de matrices globales de la estructura	43
3.3.3.1. Cálculo de matrices ampliadas de la estructura	44
3.3.4. Definición del tipo de carga aplicada a la estructura y obtención de valores y modos de la estructura	45
3.3.4.1. Cargas estáticas	45
3.3.4.2. Cargas dinámicas	46
3.3.4.3. Valores y modos propios	46
3.3.5. Obtención de las matrices reducidas	47
3.3.6. Obtención de desplazamientos	48
3.3.7. Obtención de fuerzas y reacciones	48

3.3.8. Visualización de resultados	48
<b>3.4. Interfaz</b>	49
3.4.1. Menú Principal	49
3.4.2. Definición de datos de la estructura	50
3.4.2.1. Definición del número de nodos de la estructura	51
3.4.2.2. Coordenadas de cada nodo	52
3.4.2.3. Número de barras de la estructura	53
3.4.2.4. Nodos de cada barra	54
3.4.2.5. Número de apoyos que tiene la estructura	55
3.4.2.6. Especificación del tipo de apoyo	56
3.4.2.7. Número de muelles que tiene la estructura	57
3.4.2.8. Definición de la constante de rigidez de los muelles de la estructura	58
3.4.2.9. Número de amortiguadores que tiene la estructura	59
3.4.2.10. Definición de la constante de amortiguamiento de los amortiguadores de la estructura	60
3.4.2.11. Definición de la sección de las barras de la estructura	61
3.4.2.11.1. Botón “Tipos de Sección”	61
3.4.2.11.2. Botón “Especificación Características”	62
3.4.2.11.3. Botón “Asignación Sección Barra”	60
3.4.3. Definición del tipo de fuerza aplicada	63
3.4.3.1. Botón “Número de nodos con una fuerza aplicada”	64
3.4.3.2. Botón “Indique el sentido de la fuerza”	65
3.4.3.3. Botón “Intervalo de Tiempo”	65
3.4.4. Obtención de desplazamientos y fuerzas	66
3.4.5. Obtención de valores y modos propios de la estructura	69
<b>3.5. Estructura del programa</b>	70
<b>4. Validación del programa</b>	73
<b>4.1. Estructura 1</b>	75
4.1.1. Obtención de desplazamientos y reacciones en estático	75

4.1.1.1. Resolución analítica	75
4.1.1.2. Resolución con Ed-Tridim	80
4.1.1.3. Resolución con UCMM	81
4.1.2. Valores y modos propios de la estructura	82
4.1.2.1. Cálculo analítico	82
4.1.3. Obtención de desplazamientos y reacciones en dinámico	84
4.1.3.1. Desplazamiento en el nodo tres	84
4.1.3.2. Reacción vertical del nodo dos	87
4.1.4. Introducción de muelles y amortiguadores en la estructura	89
4.1.4.1. Introducción de una muelle en la estructura	89
4.1.4.2. Introducción de un amortiguador en la estructura	91
<b>4.2. Estructura 2</b>	<b>91</b>
4.2.1. Obtención de desplazamientos en dinámico	94
4.2.2. Introducción de muelles y amortiguadores en la estructura	96
4.2.2.1. Introducción de muelles en la estructura	96
4.2.2.2. Introducción de amortiguadores en la estructura	99
<b>4.3. Estructura 3</b>	<b>101</b>
4.3.1. Obtención de desplazamientos y reacciones en dinámico	103
4.3.2. Introducción de muelle y amortiguador en la estructura	105
<b>5. Conclusiones y trabajos futuros</b>	<b>109</b>
<b>5.1. Conclusiones</b>	<b>110</b>
<b>5.2. Trabajos futuros</b>	<b>111</b>
<b>6. Bibliografía</b>	<b>112</b>
<b>7. Anexos</b>	<b>114</b>

## Índice de Figuras

Figura 1.1. Puente	2
Figura 1.2. Estructura diseñada con CypeCAD	3
Figura 2.1. Sistema de ejes locales	8
Figura 2.2. Ejes locales y globales	9
Figura 2.3. Barra de estructura reticulada	12
Figura 2.4. Reacciones	12
Figura 2.5. Distribución de cargas	14
Figura 2.6. Ejes locales en función de ejes globales	15
Figura 2.7. Ejemplo práctico	17
Figura 2.8. Ejemplo estructura	18
Figura 2.9. Escritorio de MatLab (MatLab Desktop)	21
Figura 2.10. Workspace y Variable Editor	23
Figura 2.11. Editor	24
Figura 2.12. Jerarquía gráfica de MatLab	25
Figura 2.13. Push Button y Toggle Button	27
Figura 2.14. Check Box	27
Figura 2.15. Radio Button	27
Figura 2.16. Slider	27
Figura 2.17. Listbox	28
Figura 2.18. Static Text	28
Figura 2.19. Edit Text	28
Figura 2.20. Frames	28
Figura 2.21. Ventana Guide Control Panel	29
Figura 2.22. Ventana Property Inspector	29
Figura 2.23. Panel para seleccionar el tipo de estructura	31
Figura 2.24. Barra de botones del módulo Biblioteca	31
Figura 2.25. Menú donde se elige el tipo de sección	33
Figura 2.26. Panel para la selección del tipo de apoyo	33
Figura 2.27. Barra de botones de los módulos Ejemplos y Ejercicios	34
Figura 2.28. Ventana con el dibujo de las leyes de esfuerzos	35
Figura 3.1. Primeros modos de vibración de un oscilador Simple	47
Figura 3.2. Ventana del menú principal del programa	49



Figura 3.3. Ventana definición estructura reticulada	50
Figura 3.4. Ventana número de nodos ER	51
Figura 3.5. Cuadro edición de texto del número de nodos	51
Figura 3.6. Ventana de introducción de coordenadas de cada nodo	52
Figura 3.7. Tabla con el valor de las coordenadas de cada nodo	53
Figura 3.8. Ventana para introducir el número de barras de la estructura	53
Figura 3.9. Indicación del número de barras introducido	54
Figura 3.10. Ventana para indicar el nodo de menor y mayor numeración de la barra	54
Figura 3.11. Representación de los nodos de cada barra en la ventana Definición Estructuras Reticuladas	55
Figura 3.12. Ventana para introducir el número de apoyos de la estructura	55
Figura 3.13. Verificación de datos introducidos	56
Figura 3.14. Ventana para especificar el tipo de apoyo	56
Figura 3.15. Verificación de los GDL de cada nodo	57
Figura 3.16. Ventana para introducir nº de muelles	57
Figura 3.17. Ventana en la que se indica que elemento es un muelle y su cte de rigidez	58
Figura 3.18. Tabla en la que se muestra que la barra 1 es un muelle	59
Figura 3.19. Ventana para introducir nº de amortiguadores	59
Figura 3.20. Ventana en la que se indica que elemento es un amortiguador y su cte de amortiguamiento	60
Figura 3.21. Tabla en la que se muestra que la barra 1 es un amortiguador	60
Figura 3.22. Ventana para especificar la sección de cada una de las barras	61
Figura 3.23. Ventana para introducir el nº de distintas secciones que hay en la estructura	61
Figura 3.24. Ventana para especificar las	

características de la sección	62
Figura 3.25. Ventana para asignar sección a cada barra	62
Figura 3.26. Comprobación de características de cada barra	63
Figura 3.27. Ventana donde se definirán los parámetros de la fuerza aplicada	64
Figura 3.28. Ventana para indicar el nº de nodos en que se va a aplicar una fuerza	64
Figura 3.29. Ventana de definición de los parámetros de la fuerza aplicada	65
Figura 3.30. Ventana para introducir el intervalo de tiempo	65
Figura 3.31. Ventana para introducir el intervalo de tiempo	66
Figura 3.32. Ventana para calcular y comprobar resultados	67
Figura 3.33. Ventana donde se grafican los desplazamientos frente al tiempo	67
Figura 3.34. Ventana para obtención de desplazamientos	68
Figura 4.1. Estructura articulada 1	75
Figura 4.2. Estructura para resolución mediante Ed-Tridim	80
Figura 4.3. Estructura 1 con muelle	89
Figura 4.4. Estructura 1 con muelle y amortiguador	91
Figura 4.5. Estructura dos	93
Figura 4.6. Estructura dos resuelta mediante Ed-Tridim	93
Figura 4.7. Estructura dos con muelles	96
Figura 4.8. Estructura dos con muelles y amortiguadores	99
Figura 4.9. Estructura tres	101
Figura 4.10. Resolución con Ed-Tridim	102
Figura 4.11. Estructura tres con muelle y amortiguador	105

# 1. Introducción

## **1.1. Motivación**

La sociedad ha avanzado a lo largo del tiempo gracias a la mejora del conocimiento, de la innovación, investigación y creación de técnicas destinadas a solventar las distintas necesidades o problemas que surgen.

Actualmente se vive rodeado de todas estas mejoras en las que, un alto porcentaje de la sociedad, solo admira las facilidades y mejora que suponen en sus vidas. Sin profundizar en las dificultades que su desarrollo ha supuesto y en la complejidad que incorporan.

Uno de los elementos más visibles para la sociedad, son las infraestructuras hidráulicas(presas, embalses), las infraestructuras de transportes (tanto de ferrocarril como carreteras) con sus correspondientes túneles y viaductos, los elementos de transporte (coches, trenes, aviones), y también los edificios, naves industriales o comerciales, etc.



Figura 1.1: Puente

Centrándose en la primeras etapas de creación de las infraestructuras o elementos mecánicos, se podrá observar que todas ellas tienen algo en común, y eso son las estructuras. A partir de ellas se generan los modelos de lo que se quiere crear, proporcionándose resistencia para que se puedan superar todos esos esfuerzos a los que estarán sometidos.

En la fabricación de estructuras, tanto las destinadas a la edificación, como las empleadas para otros fines, el condicionante principal es la seguridad. La estructura debe estar dimensionada para que cumpla unos coeficientes de seguridad. De forma que pueda responder a la responsabilidad que se demanda en ésta y conforme a las distintas normativas aplicables.

Como segundo objetivo, se tendrá el de cumplir las exigencias del mercado, en el que se busca precios de venta lo mas reducidos. Por otro lado, el progresivo aumento de precio de materias primas (en especial el acero), hace que se minimice el material consumido en la elaboración de las distintas estructuras para conseguir una ventaja competitiva en el mercado.

Es entre estos dos aspectos anteriores donde el cálculo de estructuras cobra una vital importancia, como una forma de poder obtener una estructura lo más competitiva posible, garantizando el correcto comportamiento que con ella se busca ante las solicitaciones a la que estará sometida.

Lo mas habitual es el estudio de deformaciones y tensiones en estructuras bajo cargas estáticas. Existe software informático destinados a este fin:

- Programas simplificados para calcular estructuras sometidas a cargas estáticas, como el Ed-Tridim. Éste permite realizar el cálculo de vigas y pórticos hiperestáticos en dos y tres dimensiones, mediante el uso de métodos matriciales y con un interfaz de usuario bastante cómodo.
- Programas basados en normativas, cuyo uso es mas complicado de usar, aunque también se analizan estructuras bajo cargas estáticas. Se puede destacar el CYPECAD, que se emplea para realizar el cálculo y dimensionamiento de viviendas, edificios y proyectos de obra civil, con aplicaciones de cargas que no varían a lo largo del tiempo, y cuyo uso, requiere un previo aprendizaje.

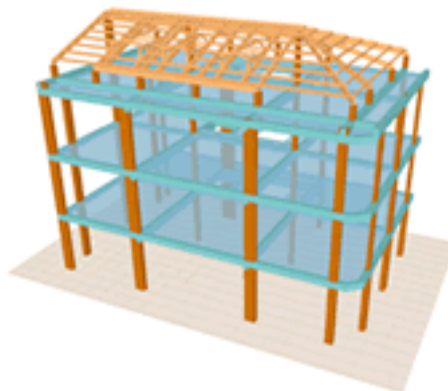


Figura 1.2: Estructura diseñada con CYPECAD.

Aunque una gran cantidad de estructuras pueden ser diseñadas correctamente considerando que las cargas fueran estáticas, existen excepciones, y es importante que el diseñador sea capaz de distinguir cuando debe aplicar cargas estáticas, y en que otros casos, cargas dinámicas.

Es importante el cálculo de estructuras bajo cargas dinámicas, en parte por los avances tecnológicos, que han hecho posible diseños mas exactos. También debido a que cada vez se buscan estructuras más ligeras y flexibles, y por tanto, mas susceptibles a efectos dinámicos. Como ejemplos:

- Estructuras en las que se aplican fuerzas súbitamente, como ráfagas de viento.
- Estructuras sometidas a fuerzas alternantes causadas por maquinaria oscilante.
- Estructuras sometidas a cargas móviles como puentes.

Para el cálculo de estructuras sometidas a cargas dinámicas, se emplean programas basados en elementos finitos (MEF), como el Abaqus o COSMOS/M, con un manejo mucho mas complejo para el usuario y que no compensa utilizar en el caso de analizar estructuras simples.

Debido a que no existe una herramienta sencilla para calcular estructuras en régimen dinámico, y con la que además se pueda añadir elementos disipativos. En este Proyecto Fin de Carrera, se ha desarrollado un programa informático destinado a ese fin, al que se ha denominado “Universal Calculus based on Matrix Method (UCMM)”.

Este programa permite realizar el cálculo de desplazamientos, fuerzas y reacciones de estructuras articuladas y reticuladas sometidas tanto a cargas dinámicas como estáticas, con un interfaz de fácil manejo y sin necesidad de que el usuario conozca al detalle toda la teoría relacionada con esta técnica. La creación de este programa se ha realizado mediante el programa MatLab y sus herramientas.

## **1.2. Objetivos**

El objetivo de este proyecto es la programación mediante MatLab de UCMM y su interfaz, que permiten obtener en estructuras articuladas y reticuladas planas, los desplazamientos, fuerzas y reacciones que se producen en éstas, cuando se las somete a cargas estáticas o dinámicas.

Los principales requisitos de este programa son:

- Evitar la redundancia en la entrada de datos para la definición del problema.
- Definición de estructuras articuladas y reticuladas.
- Análisis estáticos y dinámico de las estructuras.
- Introducción de elementos amortiguadores disipativos.
- Optimizar los cálculos.
- Modificar la estructura después de realizarse los cálculos.

- Una utilización sencilla e intuitiva del interfaz.
- Representación de resultados que permita la comprensión de los mismos.

Para su realización ha sido necesario:

- Desarrollo de un algoritmo matricial.
- Desarrollo de un algoritmo de resolución de ecuaciones diferenciales ordinarias.
- Previo aprendizaje y estudio de la programación en MatLab.
- Familiarización con el programa Ed-Tridim para realizar cálculo de estructuras.

### **1.3. Resumen**

El presente documento, consta de siete capítulos, siendo el primero de ellos la presente introducción. Se describe a continuación el contenido de cada uno de los capítulos:

- En el Capítulo II se encuentran las bases empleadas para la realización de los cálculos y su programación.
- En el Capítulo III se explican los pasos de creación del programa y modo de empleo.
- En el Capítulo IV se recogen resultados que validan la programación.
- En el Capítulo V se muestran las conclusiones, así como los posibles trabajos futuros que puede generar este proyecto.
- En el Capítulo VI recoge todas las referencias bibliográficas empleadas para la realización del trabajo.
- En el capítulo VII se recogen los anexos, en los que se muestran todas las líneas de programación de UCMM.

## 2. Antecedentes



## 2.1. Teoría del cálculo matricial

### 2.1.1 Método de la Rigidez

#### 2.1.1.1. Introducción

Un sistema estructural, constituido por un entramado de barras rectas de sección constante y que cumplen la hipótesis de pequeñas deformaciones, se puede resolver por medio de la ecuación matricial que relaciona las cargas en los nodos  $\{F\}$  y sus desplazamientos  $\{X\}$  a través de la matriz de rigidez  $[K]$  de la estructura <sup>[1]</sup>.

$$\{F\} = [K] * \{X\} \quad (2.1)$$

Para estudiar una estructura por el método de la rigidez, al igual que en cualquier otro problema elástico, se dispone de tres conjuntos de ecuaciones que deben cumplirse:

- Ecuaciones de compatibilidad .
- Ecuaciones constitutivas.
- Ecuaciones de equilibrio.

Las *Ecuaciones de compatibilidad* relacionan las deformaciones de barras con los desplazamientos nodales. Si se introducen estas relaciones en las *Ecuaciones constitutivas*, se relacionan las fuerzas en los extremos de barras con los desplazamientos nodales. Introduciéndose estas últimas relaciones en las *Ecuaciones de equilibrio* se obtiene un conjunto de ecuaciones de fuerzas nodales en función de desplazamientos nodales, que pueden ser consideradas como Ecuaciones de Equilibrio de la estructura en función de desplazamientos. La resolución de este sistema de ecuaciones permite obtener el valor de las incógnitas (desplazamientos nodales), a partir de los cuales se obtienen las solicitaciones de las barras de la estructura, así como las reacciones.

Cuando se van a calcular las relaciones esfuerzos de extremos de barra – desplazamientos, es neutral escoger un sistema de coordenadas que haga estas ecuaciones lo mas sencillas posible.

Se tomara por lo tanto como eje 'x' el que coincide con el eje geométrico de la pieza y los ejes 'y' y 'z' coincidentes con los ejes principales de la sección transversal.

Tal sistema pertenece a la barra, y no depende de la orientación de la misma en la estructura y se denominara sistema de ejes locales.

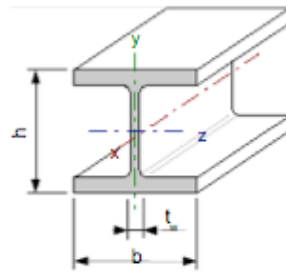


Figura 2.1: Sistema de ejes locales.

Por el contrario, cuando las piezas se unen entre sí para formar la estructura, es necesario tener un sistema de coordenadas común para todos los movimientos y esfuerzos de extremo de barras para poder aplicar las condiciones de equilibrio y compatibilidad. A dicho sistema se lo denominará como sistema de ejes globales.

Dichos esfuerzos de extremos de barras y desplazamientos dependerán del tipo de estructura que se esté resolviendo, para barras de:

- (a) Estructura Articulada: dos desplazamientos por nudo (desplazamiento horizontal y vertical).
- (b) Estructura Reticulada: tres desplazamientos por nudo (desplazamiento horizontal, vertical y giro).

### 2.1.1.2. Procedimiento para el cálculo matricial de estructuras

Se pueden considerar seis etapas fundamentales en la resolución de un problema:

- (1) Identificación estructural.
- (2) Cálculo de la matriz de rigidez de barra y del vector de cargas nodales equivalente.
- (3) Rotación de ejes en el plano.
- (4) Matriz de rigidez global de estructura.
- (5) Matriz de cargas globales de la estructura.
- (6) Condiciones de contorno y cálculo de reacciones.

### 2.1.1.2.1. Identificación estructural

Consiste en definir a través de números y datos las barras de la estructura.

- Definir un sistema de ejes globales para la estructura. Las coordenadas de los nodos se refieren a dicha sistema.
- Conectividad de los elementos, se identifica para cada barra el nodo inicial y final. La misma queda definida automáticamente por el orden establecido para la numeración de los nodos de la barra. El eje 'x' local coincide con el eje geométrico de la barra, siendo el sentido positivo el que va del nodo de menor numeración al de mayor numeración. Los otros ejes formaran un triedro directo.

### 2.1.1.2.2. Matriz de rigidez y vector de cargas nodales equivalente

- (a) *Barra de estructura articulada*: se considera una barra de una estructura articulada, suponiendo que la misma esta orientada con relación a un sistema de ejes globales 'x' e 'y'.

Se supondrá que la barra es recta, de sección transversal constante y que el material responde a la ley de Hooke.

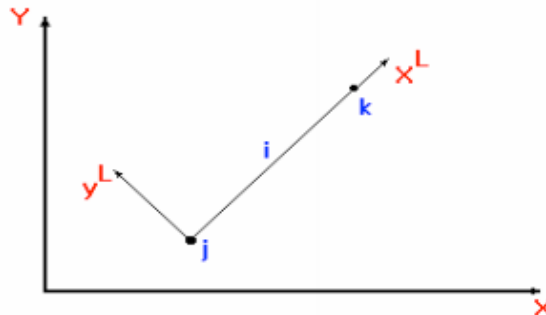


Figura 2.2: Ejes locales y globales.

En la barra 'i' de la figura, el nodo inicial es el 'j' y el final es el 'k', quedando definida la orientación de la misma por los ejes locales 'x' e 'y'.

Se considera que no existen deformaciones iniciales y que la deformación es elástica. En este caso el alargamiento de la barra 'i' estará dado por:

$$\Delta L_i = X_{X_k}^L - X_{X_j}^L \quad (2.2)$$

Donde  $X_{X_k}^L$  y  $X_{X_j}^L$  son los desplazamientos del nudo 'k' y 'j' respectivamente en la dirección local 'x<sup>L</sup>'.

Para una barra de una estructura articulada existe una sola sollicitación posible que es el esfuerzo axial o normal. Suponiendo un material elástico lineal sometido a esfuerzo de tracción tendremos para los nudos 'j' y 'k' respectivamente:

$$\begin{aligned} F_{X_k} &= \frac{E \cdot A}{L} \Delta L_i \rightarrow \frac{E \cdot A}{L} (X_{X_k}^L - X_{X_j}^L) \\ F_{X_j} &= -\frac{E \cdot A}{L} \Delta L_i \rightarrow -\frac{E \cdot A}{L} (X_{X_k}^L - X_{X_j}^L) \end{aligned} \quad (2.3)$$

Donde:

E → Modulo de elasticidad.

L → Longitud de la barra.

A → Área de la sección transversal de la barra.

Como en la dirección 'y<sup>L</sup>' para barras de estructuras articuladas no existen sollicitaciones se pueden expresar las ecuaciones anteriores en forma matricial:

$$\begin{bmatrix} F_{X_j} \\ F_{Y_j} \\ F_{X_k} \\ F_{Y_k} \end{bmatrix} = \begin{bmatrix} \frac{E \cdot A}{L} & 0 & -\frac{E \cdot A}{L} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{E \cdot A}{L} & 0 & \frac{E \cdot A}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_{X_j}^L \\ X_{Y_j}^L \\ X_{X_k}^L \\ X_{Y_k}^L \end{bmatrix} \quad (2.4)$$

La expresión anterior se corresponde a la ecuación matricial de la barra 'i' en coordenadas locales y expresa las fuerzas de extremo de barra FI<sup>L</sup> en función de los desplazamientos de los nodos XI<sup>L</sup>.

A la matriz que relaciona FI<sup>L</sup> y XI<sup>L</sup> se la denomina matriz de rigidez de barra de la estructura articulada en coordenadas locales KI<sup>L</sup>. Expresado en forma compacta o simbólica:

$$FI^L = KI^L * XI^L \quad (2.5)$$

La ecuación (2.5) define las fuerzas de extremo F<sub>j</sub> y F<sub>k</sub> para cualquier pareja de desplazamientos X<sub>j</sub>, X<sub>k</sub> dados. Estas ecuaciones son simétricas, como se podía esperar a partir del teorema de reciprocidad. No es posible, sin embargo, resolverlas y obtener los desplazamientos (X) en términos de las fuerzas (F), puesto que la matriz K es singular. Esto refleja el hecho de que la pieza puede

sufrir un movimiento arbitrario de conjunto, sin afectar las fuerzas que actúan en sus extremos.

Para un muelle sometido a esfuerzo de tracción se tendrá para los nudos 'j' y 'k' respectivamente:

$$\begin{aligned} F_{X_k} &= K \cdot \Delta L_i \rightarrow K \cdot (X_{X_k}^L - X_{X_j}^L) \\ F_{Y_k} &= K \cdot \Delta L_i \rightarrow K \cdot (X_{X_k}^L - X_{X_j}^L) \end{aligned} \quad (2.6)$$

Donde:

$K \rightarrow$  Constante de rigidez del muelle.

$$\begin{bmatrix} F_{X_j} \\ F_{Y_j} \\ F_{X_k} \\ F_{Y_k} \end{bmatrix} = \begin{bmatrix} K & 0 & K & 0 \\ 0 & 0 & 0 & 0 \\ K & 0 & K & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_{X_j}^L \\ X_{Y_j}^L \\ X_{X_k}^L \\ X_{Y_k}^L \end{bmatrix} \quad (2.7)$$

Para un amortiguador sometido a esfuerzo de tracción se tendrá para los nudos 'j' y 'k' respectivamente:

$$\begin{aligned} F_{X_k} &= C \cdot (\dot{X}_{X_k}^L - \dot{X}_{X_j}^L) \\ F_{Y_k} &= C \cdot (\dot{X}_{X_k}^L - \dot{X}_{X_j}^L) \end{aligned} \quad (2.8)$$

Donde:

$C \rightarrow$  Constante de amortiguamiento del amortiguador.

$$\begin{bmatrix} F_{Xj} \\ F_{Yj} \\ F_{Xk} \\ F_{Yk} \end{bmatrix} = \begin{bmatrix} C & 0 & C & 0 \\ 0 & 0 & 0 & 0 \\ C & 0 & C & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} \dot{X}_{Xj}^L \\ \dot{X}_{Yj}^L \\ \dot{X}_{Xk}^L \\ \dot{X}_{Yk}^L \end{bmatrix} \quad (2.9)$$

(b) Barra de estructura reticulada: para este tipo de elemento se tendrán tres desplazamientos (dos traslaciones y una rotación en el plano).

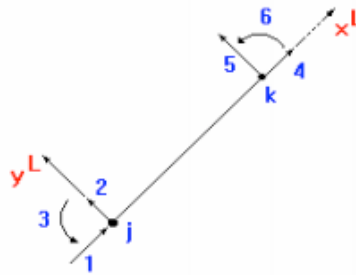


Figura 2.3: Barra de estructura reticulada <sup>[1]</sup>.

La matriz de rigidez se obtiene dando desplazamientos unitarios de a uno por vez en las direcciones de la figura mientras los otros permanecen nulos.

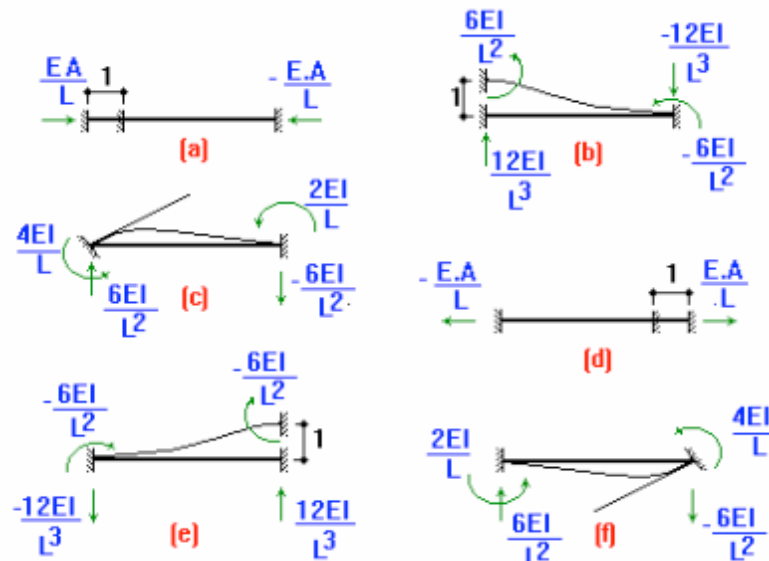


Figura 2.4: Reacciones <sup>[1]</sup>

Las reacciones mostradas en la figura anterior constituyen las respectivas columnas de la matriz de rigidez de la barra de una estructura reticulada de la ecuación (2.10) :

$$\begin{bmatrix} F_{Xj} \\ F_{Yj} \\ F_{Zj} \\ F_{Xk} \\ F_{Yk} \\ F_{Zk} \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \cdot \begin{bmatrix} X_{Xj}^L \\ X_{Yj}^L \\ X_{Zj}^L \\ X_{Xk}^L \\ X_{Yk}^L \\ X_{Zk}^L \end{bmatrix} \quad (2.10)$$

Escrito en forma compacta:

$$F^L = K^L * X^L \quad (2.11)$$

Esta matriz relaciona las fuerzas de extremo de barra con los desplazamientos nodales en ejes locales.

Para un muelle se tendrá:

$$\begin{bmatrix} F_{Xj} \\ F_{Yj} \\ F_{Zj} \\ F_{Xk} \\ F_{Yk} \\ F_{Zk} \end{bmatrix} = \begin{bmatrix} K & 0 & 0 & K & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K & 0 & 0 & K & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_{Xj}^L \\ X_{Yj}^L \\ X_{Zj}^L \\ X_{Xk}^L \\ X_{Yk}^L \\ X_{Zk}^L \end{bmatrix} \quad (2.12)$$

Y para un amortiguador:

$$\begin{bmatrix} F_{X_j} \\ F_{Y_j} \\ F_{Z_j} \\ F_{X_k} \\ F_{Y_k} \\ F_{Z_k} \end{bmatrix} = \begin{bmatrix} C & 0 & 0 & C & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & C & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{X}_{X_j}^L \\ \dot{X}_{Y_j}^L \\ \dot{X}_{Z_j}^L \\ \dot{X}_{X_k}^L \\ \dot{X}_{Y_k}^L \\ \dot{X}_{Z_k}^L \end{bmatrix} \quad (2.13)$$

### 2.1.1.2.3. Cargas Nodales

Hasta ahora se ha supuesto que las cargas estaban aplicadas en los nodos, y por lo tanto existe una correspondencia biunívoca entre los puntos de aplicación de las cargas y los desplazamientos que están siendo calculados. Si esto no ocurriera, por ejemplo se tuviesen cargas en el tramo de las barras, en forma distribuida o concentrada, se debe sustituir las cargas en las mismas por un sistema de cargas equivalentes aplicadas en los nodos que produzca en la estructura el mismo efecto que las cargas originales.

Aplicando el principio de superposición, que es válido por haber supuesto que el sistema es lineal, se puede descomponer las cargas tal como se indica en la figura:

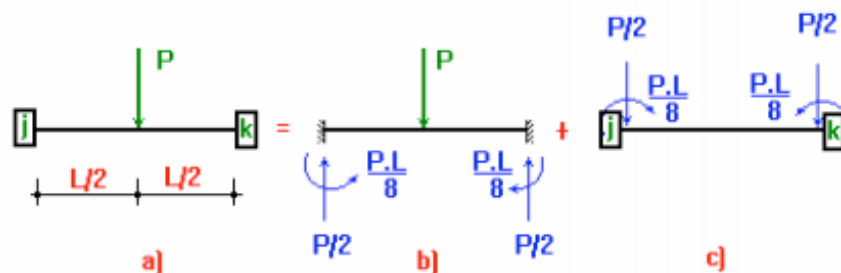


Figura 2.5: Distribución de cargas <sup>[1]</sup>

Como se puede observar las cargas, reacciones y deformaciones de la estructura a) serán equivalentes a la suma de los dos estados b) y c).

Como las deformaciones de nodos en b) son nulas, serán iguales las deformaciones de los casos c) y a). O sea que las cargas de c) producen la misma respuesta estructural en lo referente a desplazamientos de nudos que las cargas originales. Estas serán entonces las cargas equivalentes en los



nodos, que no son mas que las reacciones de empotramiento perfecto cambiadas de signo.

Los esfuerzos en los extremos de barra se obtienen por la suma de los casos (b) y (c).

$$F^a = F^b + F^c \quad D^a = D^c \quad (2.14)$$

Por lo tanto a la ecuación (2.4) habrá que añadirle las fuerza de empotramiento perfecto del caso b).

$$F^L = K^L * X^L + A^L \quad (2.15)$$

$A^L$  representa el vector de fuerzas del empotramiento perfecto de la barra en coordenadas locales.

#### 2.1.1.2.4. Rotación de ejes en el plano

Hasta el momento se representan las matrices de rigidez o amortiguamiento de los elementos de estructuras, tanto articuladas como de reticuladas, según un sistema de ejes locales, estando los desplazamientos y esfuerzos de extremo de barra referidos a los mismos.

No obstante, como ya se había mencionado, antes de poder aplicar las condiciones de equilibrio en los nodos de compatibilidad de desplazamientos es necesario transformar las fuerzas y desplazamientos en un sistema de ejes globales.

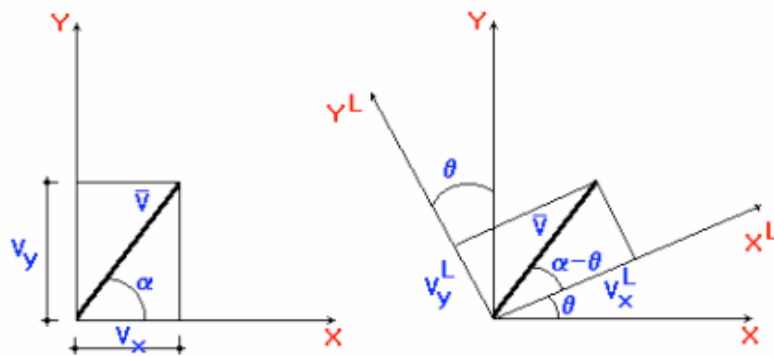


Figura 2.6: Ejes locales en función de los ejes globales <sup>[1]</sup>

Se supone el vector V de la figura referido al sistema de ejes 'x' e 'y'. Las componentes del mismo serán:

$$V_x = V * \cos \alpha \quad || \quad V_y = V * \sin \alpha \quad (2.16)$$

En el sistema de ejes 'x<sup>L</sup>' e 'y<sup>L</sup>', las componentes serán:

$$V_{x^L} = V * \cos (\alpha - \theta) \quad || \quad V_{y^L} = V * \sin (\alpha - \theta) \quad (2.17)$$

Luego:

$$V_x^L = V \cdot \cos \alpha \cdot \cos \theta + V \cdot \sin \alpha \cdot \sin \theta \quad (2.18)$$

$$V_y^L = V \cdot \cos \alpha \cdot \cos \theta + V \cdot \sin \alpha \cdot \sin \theta \quad (2.19)$$

Teniéndose en cuenta las ecuaciones (2.18) y (2.19), y escribiéndose en forma matricial:

$$\begin{bmatrix} V_x^L \\ V_y^L \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} V_x \\ V_y \end{bmatrix} \quad (2.20)$$

O en forma compacta:

$$V^L = R \cdot V \quad (2.21)$$

A la matriz de cosenos directores 'R' se la llamara matriz de rotación.

Para expresar las matrices de rigidez y amortiguamiento en ejes globales, se mide el ángulo que forma el eje 'x<sup>G</sup>' global positivo con el eje 'x<sup>L</sup>' local positivo, en sentido anti horario.

Para pasar las matrices de rigidez y amortiguamiento del sistema que se encuentra en ejes locales a ejes globales, se tendrá que multiplicar la matriz de rigidez y amortiguamiento en ejes locales (K<sup>L</sup> y C<sup>L</sup>) por la matriz de rotación (R) y a su vez por la matriz de rotación traspuesta (R<sup>t</sup>), es decir:

$$[K^G] = [R] \cdot [K^L] \cdot [R^t] \quad (2.22)$$

Para una barra de una estructura reticulada se tendrá la siguiente matriz de rotación:

$$R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

#### 2.1.1.2.5. Matriz de rigidez global de la estructura

La matrices de rigidez y amortiguamiento global de la estructura tendrá tantas filas y tantas columnas como nodos por sus grados de libertad tenga la estructura.

Se obtiene de calcular las submatrices de cada barra en la posición correspondiente . En las posiciones en las que se tenga varias submatrices se producirá el ensamble de la matriz, y a ello lo llamaremos elementos ensamblados.

Así para la siguiente estructura articulada:

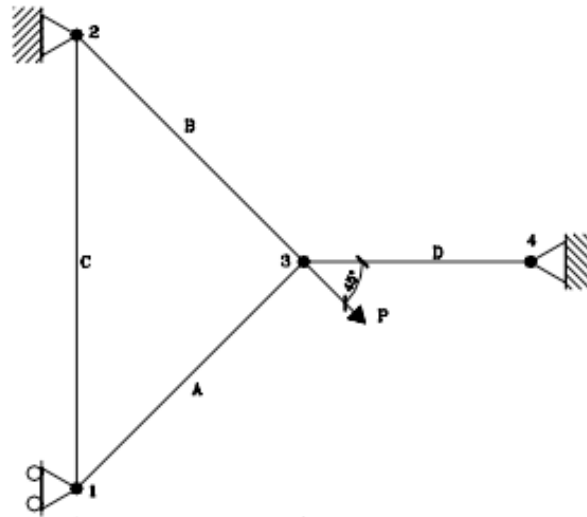


Figura 2.7: Ejemplo práctico <sup>[2]</sup>

Se pueden distinguir 4 barras:

Barra A, cuyo nodo menor es el 1 y el nodo mayor es el 3

Barra B, cuyo nodo menor es el 2 y el nodo mayor es el 3

Barra C, cuyo nodo menor es el 1 y el nodo mayor es el 2

Barra D, cuyo nodo menor es el 3 y el nodo mayor es el 4

Para cada barra se tendrá una matriz de rigidez que ocupara una posición en la matriz de rigidez de la estructura:

Barra A

$$\begin{bmatrix} (K_{11}^A)_{11} & (K_{12}^A)_{13} \\ (K_{21}^A)_{31} & (K_{22}^A)_{33} \end{bmatrix}$$

Barra C

$$\begin{bmatrix} (K_{11}^C)_{11} & (K_{12}^C)_{12} \\ (K_{21}^C)_{21} & (K_{22}^C)_{22} \end{bmatrix}$$

Barra B

$$\begin{bmatrix} (K_{11}^B)_{22} & (K_{12}^B)_{23} \\ (K_{21}^B)_{32} & (K_{22}^B)_{33} \end{bmatrix}$$

Barra D

$$\begin{bmatrix} (K_{11}^D)_{33} & (K_{12}^D)_{34} \\ (K_{21}^D)_{43} & (K_{22}^D)_{44} \end{bmatrix}$$

Cada una de las submatrices, irán ubicadas en la matriz de rigidez global de la estructura según la posición que marca el subíndice que aparece fuera del paréntesis. Y aquellas submatrices que tengan el mismo subíndice, se sumaran en la matriz de rigidez de la estructura.

Obteniéndose la siguiente matriz de rigidez de la estructura, en este caso en ejes locales:

$$\begin{bmatrix} (K_{11}^A)_{11} + (K_{11}^C)_{11} & (K_{12}^C)_{12} & (K_{12}^A)_{13} & 0 \\ (K_{21}^C)_{21} & (K_{11}^B)_{22} + (K_{22}^C)_{22} & (K_{12}^B)_{23} & 0 \\ (K_{21}^A)_{31} & (K_{21}^B)_{32} & (K_{22}^A)_{33} + (K_{22}^B)_{33} + (K_{11}^D)_{33} & (K_{12}^D)_{34} \\ 0 & 0 & (K_{21}^D)_{43} & (K_{22}^D)_{44} \end{bmatrix}$$

#### 2.1.1.2.6. Matriz de cargas global de la estructura

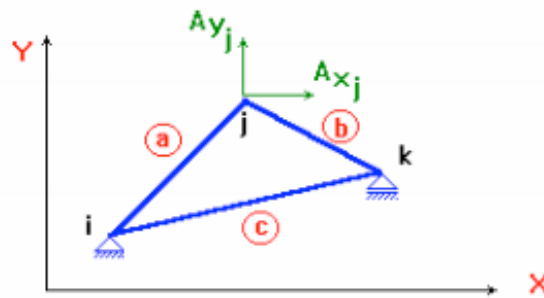


Figura 2.8: Ejemplo estructura <sup>[1]</sup>

Las ecuaciones de equilibrio exigen que las cargas externas aplicadas en los nodos deben ser iguales a la suma de las sollicitaciones de extremo de las barras que concurren al nodo.

Siendo el vector de cargas externas aplicadas en j:

$$A_j = \begin{bmatrix} A_{xj} \\ A_{yj} \end{bmatrix} \quad (2.24)$$

Centrándose únicamente en la barra (a), su ecuación matricial será:

$$F^a = K^a \cdot X^a + A^a \quad \rightarrow \quad F^a - A^a = K^a \cdot X^a \quad (2.25)$$

Donde:

$F^a \rightarrow$  sollicitación en el extremo 'i' de la barra (a)

$X^a \rightarrow$  deformación en el extremo 'i' de la barra (a)

$A^a \rightarrow$  fuerza de empotramiento perfecto en el extremo 'i' de la barra (a)

$K^a \rightarrow$  matriz de rigidez de la barra (a)

#### 2.1.1.2.7. Condiciones de contorno o de borde y cálculo de reacciones en la estructura.

Un sistema de ecuaciones:  $A = K \cdot X$  correspondiente a una estructura completa antes de aplicarse las condiciones de contorno es indeterminado, pues  $K$  es singular. La razón de esta singularidad es el resultado de no haberse considerado las vinculaciones o apoyos de la estructura en el exterior. Al introducirse las condiciones de vínculo desaparece la indeterminación (o singularidad de la matriz de rigidez), siempre que el número de vínculos sea por lo menos el mínimo necesario para eliminar los movimientos de cuerpo rígido de la estructura.

El conocimiento de determinados movimientos nodales, disminuye el número de incógnitas, tornándose innecesarias las ecuaciones correspondientes a estos corrimientos.

La eliminación de la ecuación de un desplazamiento implica la destrucción de la banda de la matriz, lo cual exigirá un reacomodamiento de las incógnitas del problema.

Se puede expresar el sistema de ecuaciones  $A = K \cdot X$ , de la siguiente forma:

$$\begin{bmatrix} A_H \\ A_V \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ X_V \end{bmatrix} \quad (2.26)$$

Donde:

$A_H \rightarrow$  reacciones de apoyo con desplazamiento impedido ( $X_H = 0$ ), reacciones que por el momento se desconoce.

$A_V \rightarrow$  agrupa las fuerzas externas sobre nodos con desplazamientos  $X_V$  desconocidos.

$K \rightarrow$  matriz de rigidez que relaciona fuerzas conocidas (cargas externas  $A_V$ ) con desplazamientos desconocidos  $X_V$ .

Por lo que solo se tendrá que plantear las ecuaciones y obtener el desplazamiento, a lo que se podría llamar sistemas de ecuaciones reducidas:

$$A_V = K_{22} \cdot X_V \quad (2.27)$$

Obtenidas las incógnitas de los desplazamientos en la estructura, el siguiente paso sería la obtención de las reacciones de la estructura, planteándose las ecuaciones que obtendríamos de las siguientes matrices:

$$F - A = K \cdot X \quad (2.28)$$

Y despejando F, que proporcionaría los valores de las reacciones de la estructura.

## **2.2. MatLab**

### **2.2.1. Introducción**

MatLab es el nombre abreviado de “MATrix LABoratory” [1]. MatLab con un lenguaje de programación propio, consiste en un programa que realiza cálculos numéricos con vectores y matrices. Como caso particular puede trabajar también con escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información mas complejas. Una de las capacidades mas atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MatLab dispone de un código básico y de varias librerías especializadas (toolboxes).

### **2.2.2. Entorno de trabajo de Matlab**

Se trata de un entorno de trabajo muy gráfico e intuitivo , cuyas componentes mas importante son:

1. El escritorio de MatLab (MatLab Desktop), que es la ventana o contenedor de máximo nivel en la que se pueden situar los demás componentes.
2. Las componentes individuales, orientadas a tareas concretas dentro del programa, entre las que se puede citar:
  - a. La ventana de comandos (Command Window).
  - b. La ventana histórica de comandos (Command History).
  - c. El espacio de trabajo (Workspace).
  - d. La plataforma de lanzamiento (Launch Pad).

- e. El directorio actual (Current Directory o Current Folder).
- f. La ventana de ayuda (Help).
- g. El editor de ficheros y depurador de errores (Editor&Debugger).
- h. El editor de vectores y matrices (Array Editor).
- i. La ventana que permite estudiar cómo se emplea el tiempo de ejecución (Profiler).

### 2.2.2.1. Escritorio de MatLab (MatLab Desktop)

Se trata de la ventana más general de la aplicación. El resto de ventanas o componentes citadas pueden alojarse en el escritorio de MatLab o ejecutarse como ventanas independientes.

Cuando se arranca MatLab, aparece una ventana como la mostrada en la Figura 9, aunque esta puede variar según las preferencias del usuario, quién podrá eliminar algunas de las componentes visibles y/o añadir otras. La configuración adoptada por el usuario se mantendrá la siguiente vez que arranque el programa. Es posible también guardar distintas configuraciones con distintos nombres para su uso posterior.

En el caso de la configuración mostrada en la Figura 9, podemos ver que éste escritorio de Matlab esta formado por cuatro ventanas que son el Command Window, Command History, Workspace y Current Folder.

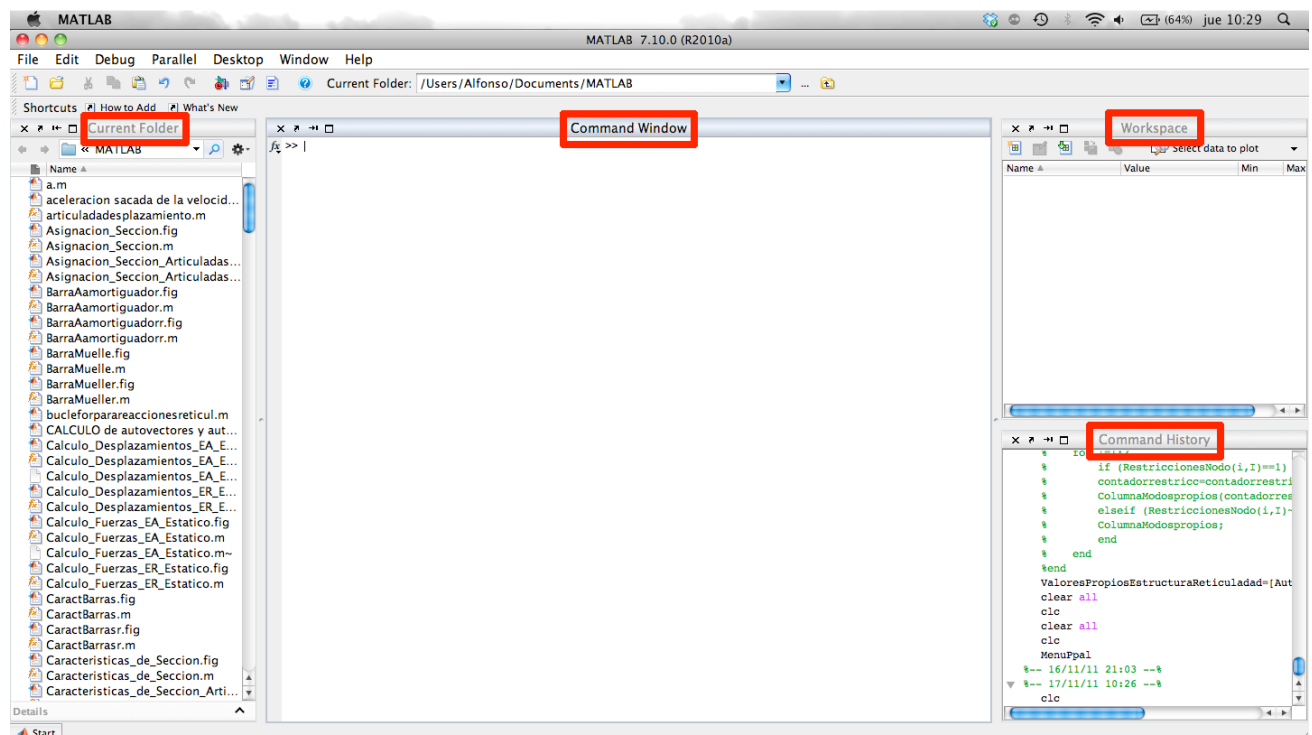


Figura 2.9: Escritorio de Matlab (Matlab Desktop)

### 2.2.2.2. Command Window

En cierta forma es la ventana mas importante. Es la que ocupa la posición central de la figura 9, en la que se ejecutan interactivamente las instrucciones de MatLab, tanto archivos \*.m como líneas escritas directamente sobre la misma y en donde se muestran los resultados correspondientes, si es el caso.

### 2.2.2.3. Command History Browser

Ocupa la posición derecha inferior de la Figura 9 y ofrece acceso a las sentencias que se han ejecutado anteriormente en el Command Window. Éstas sentencias están también disponibles mediante las teclas ↑ y ↓.

Las sentencias ejecutadas anteriormente se pueden volver a ejecutar mediante un doble clic o por medio del menú contextual que se abre al clicar sobre ellas con el botón derecho.

### 2.2.2.4. Current Directory Browser o Current Folder

Los programas MatLab se encuentran en ficheros con la extensión \*.m. Estos ficheros se ejecutan tecleando su nombre en la línea de comandos (sin la extensión), seguido de los argumentos entre paréntesis, si se trata de funciones. No todos los ficheros \*.m son accesibles sin más. Para que un fichero \*.m pueda ser ejecutado es necesario que se cumpla una de las siguientes condiciones:

1. Que esté en la Current Folder (ventana en la zona izquierda de la Figura 9) . MatLab mantiene en todo momento un único directorio con esta condición y es el primer sitio donde MatLab busca cuando desde la línea de comandos se le pide que ejecute un fichero.
2. Que esté en uno de los directorios indicados en el Path de MatLab. El Path es una lista ordenada de directorios en los que el programa busca los ficheros o las funciones que ha de ejecutar.

La ventana Current Folder permite explorar los directorios del ordenador, mostrando los ficheros deseados. Los ficheros \*.m mostrados en la ventana Current Folder se pueden abrir con el Editor/Debugger mediante un doble clic.

A partir del menú contextual que se abre clicando con el botón derecho en cualquier parte de la ventana Current Folder se tiene la posibilidad de añadir ese directorio al Path de MatLab.



### 2.2.2.5. Workspace y Array Editor o Variable Editor

El Workspace que aparece en la zona derecha superior de la Figura 9, muestra el conjunto de variables y de funciones de usuario que en un determinado momento están definidas en la memoria del programa o de la función que se esta ejecutando.

La ventana Workspace constituye un entorno gráfico para ver las variables definidas en el espacio de trabajo. Se activa con el comando View/Workspace. Y clicando dos veces sobre alguna de las variables ya predefinidas, se abre la ventana Variable Editor que permite modificar esos resultados o variables predefinidas, como se puede observar en la figura 10.

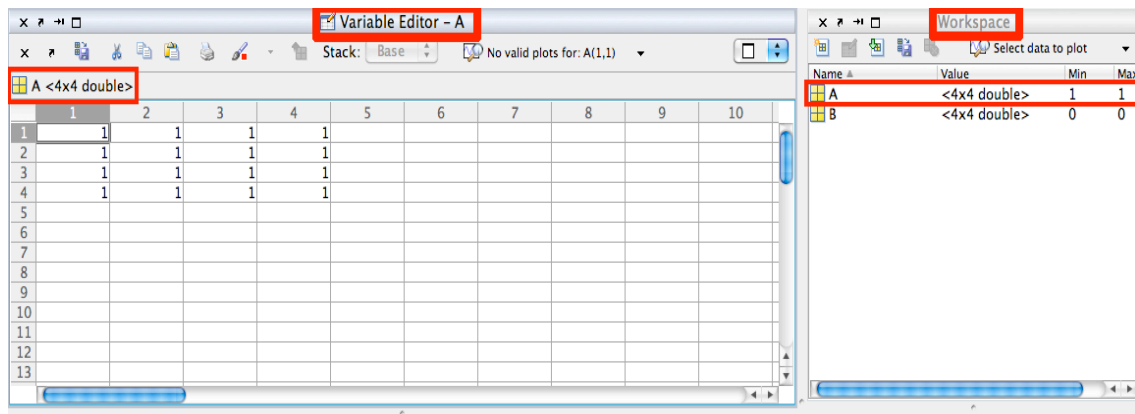


Figura 2.10: Workspace y Variable Editor.

### 2.2.2.6. Editor/Debugger

En MatLab tienen particular importancia los ya citados ficheros-M (o M-files). Son ficheros de texto ASCII, con la extensión \*.m, que contienen conjuntos de comandos o definición de funciones. La importancia de estos ficheros-M es que al teclear su nombre en la línea de comandos y pulsar Intro, se ejecutan uno tras otro los comandos contenidos en dicho fichero. El poder guardar instrucciones y grandes matrices en un fichero permite ahorrar mucho trabajo de teclado.

MatLab dispone de un editor que permite tanto crear y modificar estos ficheros, como ejecutarlos paso a paso para ver si contienen errores (proceso de Debug o depuración).

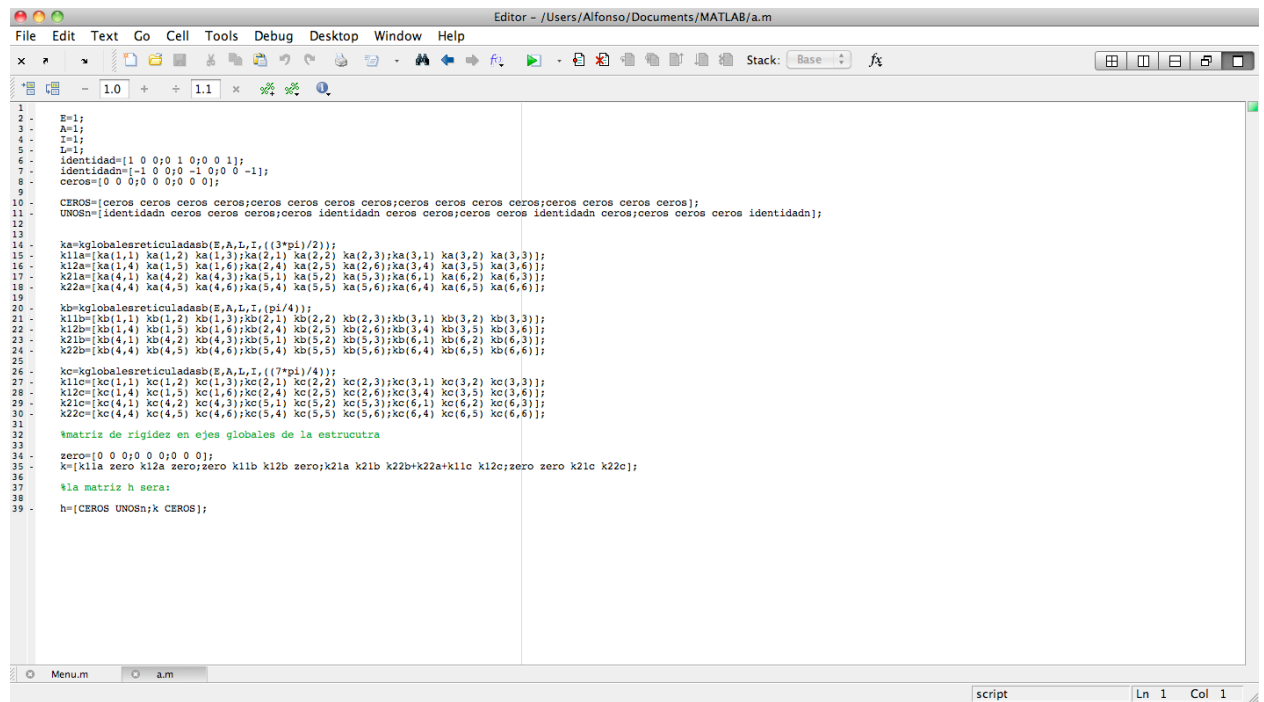


Figura 2.11: Editor

Las líneas negras, son las líneas de comandos que se van a ejecutar, mientras que las líneas verdes, son comentarios que pueden introducirse siempre precedidos del símbolo %.

### 2.2.3. Interfaz gráfica GUIDE

GUIDE (Graphical User Interface Development Environment) es un juego de herramientas que se extiende por completo en el soporte Matlab, diseñadas para crear GUIs (Graphical User Interfaces) de formas bastante sencilla e intuitiva [3].

Para poder hacer programas que utilicen las capacidades gráficas avanzadas de Matlab hay que conocer algunos conceptos que se explican en los apartados siguientes.

### 2.2.3.1. Estructura de los gráficos de MatLab

Los gráficos de MatLab tienen una estructura jerárquica formada por objetos de distintos tipos. Esta jerarquía tiene forma de árbol, con el aspecto mostrado en la figura 12.

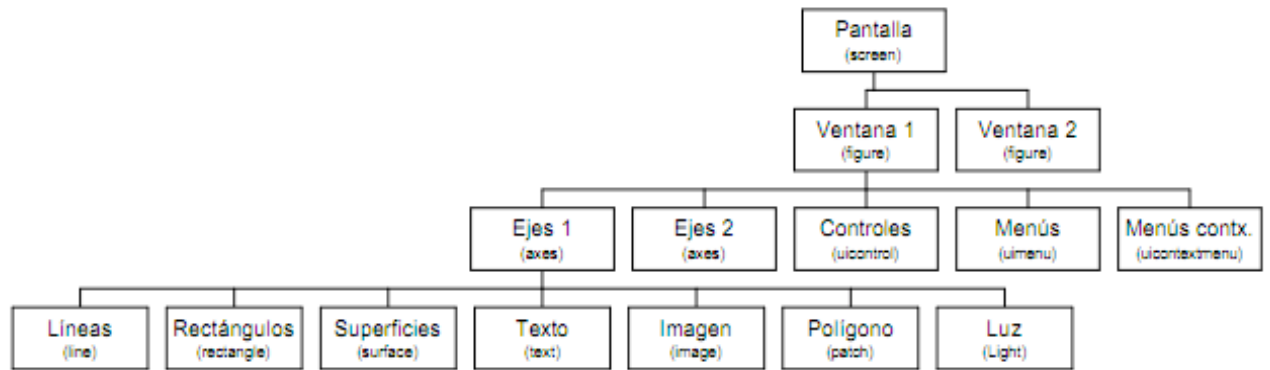


Figura 2.12: Jerarquía gráfica de MatLab

#### 2.2.3.1.1. Objetos gráficos de MatLab

Según se muestra en la Figura 12, el objeto mas general es la pantalla (screen). Este objeto es la raíz de todas las demás y solo puede haber un objeto pantalla. Una pantalla puede contener una o mas ventanas (figures). A su vez cada una de las ventanas puede tener uno o mas ejes de coordenadas (axes) en los que representar objetos de mas bajo nivel. Una ventana puede tener también controles (uicontrol) tales como botones, barras de desplazamiento, botones de selección o de opción, etc. Y menus (uimenu). Finalmente los ejes pueden contener los seis tipos de elementos gráficos que permite MatLab: líneas (line), rectángulos (rectangle), polígonos (patches), superficies (surface), imágenes bitmap (image) y texto (text).

#### 2.2.3.1.2. Identificadores (Handles)

Cada uno de los objetos de MatLab tiene un identificador único (handle).

### 2.2.3.2. Propiedades de los objetos

Todos los objetos de MatLab tienen distintas, como mismas propiedades. Algunas de las propiedades comunes a los objetos son: children, clipping, parent, type, User Data, Visible y Tag. Las propiedades tienen valores por defecto que se usan, a no ser que el usuario quiera modificarlas.

### 2.2.3.2.1. Funciones set y get

MatLab dispone de estas dos funciones que permiten consultar y cambiar el valor de las propiedades de un objeto. La función set se encarga de mostrar en pantalla lo que al usuario le interese. Mientras que la función get, permite almacenar la información introducida por el usuario.

### 2.2.3.3. Creación de controles gráficos: Comando *uicontrol*

El formato uicontrol permite definir los controles gráficos para manejar un programa. Algunas de las propiedades del comando uicontrol:

- Color de objeto (BackgroundColor) : controla el color del objeto, por defecto éste será gris claro.
- Acción a efectuar por el comando (Callback): este comando especifica la acción a efectuar por MatLab al actuar sobre el control.
- Control activado/desactivado (Enable): este comando permite desactivar un control, de tal forma que una acción sobre el mismo (por ejemplo, apretar con el ratón sobre un PushButton) no produce ningún efecto. Los valores que puede tomar esta variable son on y off.
- Alineamiento horizontal de título (HorizontalAlignment): esta opción determina la posición del título del control en el mismo (propiedad *String*).
- Valor máximo (Max): determina el máximo valor que puede tomar la propiedad *Value* cuando se utilizan cajas de textos, botones o barras de desplazamiento.
- Valor mínimo (Min): análogo a la opción anterior pero para el valor mínimo.
- Identificador del objeto padre (Tag): esta opción especifica el tag del objeto padre. Uno de los elementos mas importantes.
- Posición del objeto (Position): en esta opción se determina la posición y el tamaño del control del objeto padre.
- Nombre del objeto (String): esta opción define el nombre que aparecerá en el control.
- Unidades (Units): unidades de medida en las que se interpretara el comando *Position*.
- Visible (Visible): indica si el control es visible en la ventana y puede tomar dos valores: on/off.

### 2.2.3.4. Tipos de *uicontrol*

Existen ocho tipos de controles diferentes. La utilización de cada uno vendrá dada en función de sus características y aplicación:

1. Botones (Push Buttons y Toggle Buttons): al clicar sobre el con el ratón se producirá un evento que lanza una acción que deberá ser ejecutada por MatLab.



Figura 2.13: Push Button y Toggle Button

2. Botones de selección (Check Boxes): los botones de selección permiten al usuario seleccionar entre las distintas opciones que aparezcan en la misma.

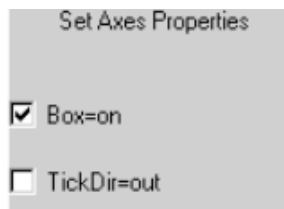


Figura 2.14: Check Box

3. Botones de opción (Radio Buttons): al igual que los botones de selección, los botones de opción permiten al usuario seleccionar entre varias posibilidades. La diferencia reside en que los botones de opción son excluyentes, es decir no puede haber mas de uno activado, mientras que el control anterior permite tener dos o más cajas activadas.

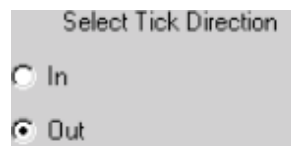


Figura 2.15: Radio Buttons

4. Barras de desplazamiento (Slider): permiten al usuario introducir un valor entre un rango de valores, que irán variando según se posicione la barra.



Figura 2.16: Slider

5. Cajas de selección (Listbox): permite elegir una opción entre varias mostradas en una lista.

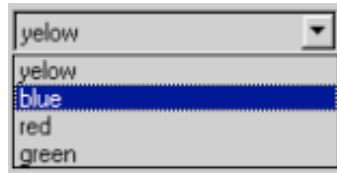


Figura 2.17: Listbox

6. Cajas de texto (Static Text): son controles espaciales, ya que no permiten realizar ninguna operación con el ratón. Permite escribir un texto en la pantalla de modo informativo.

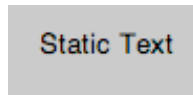


Figura 2.18: Static Text

7. Cajas de texto editables (Edit Text): se emplean para introducir y modificar cadenas de caracteres, e incluso para mostrar datos al usuario. Su función editable dependerá de si su Enable se encuentra en estado On u Off.

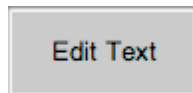


Figura 2.19: Edit Text

8. Marcos (Frames): no es un control propiamente dicho. Su función es la de englobar un serie de opciones (botones, cajas de texto, etc.) con el fin de mantener una estructura ordenada de controles.

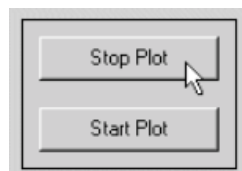


Figura 2.20: Frames

### 2.2.3.5. Construcción de una interfaz gráfica

Para ejecutar GUIDE, se escribirá en la línea de comandos de MatLab :

```
>>guide
```

y pulsando *Intro*. O bien, seleccionando File→New→GUI. Automáticamente se abrirá una pantalla como la que se muestra en la figura 2.21:

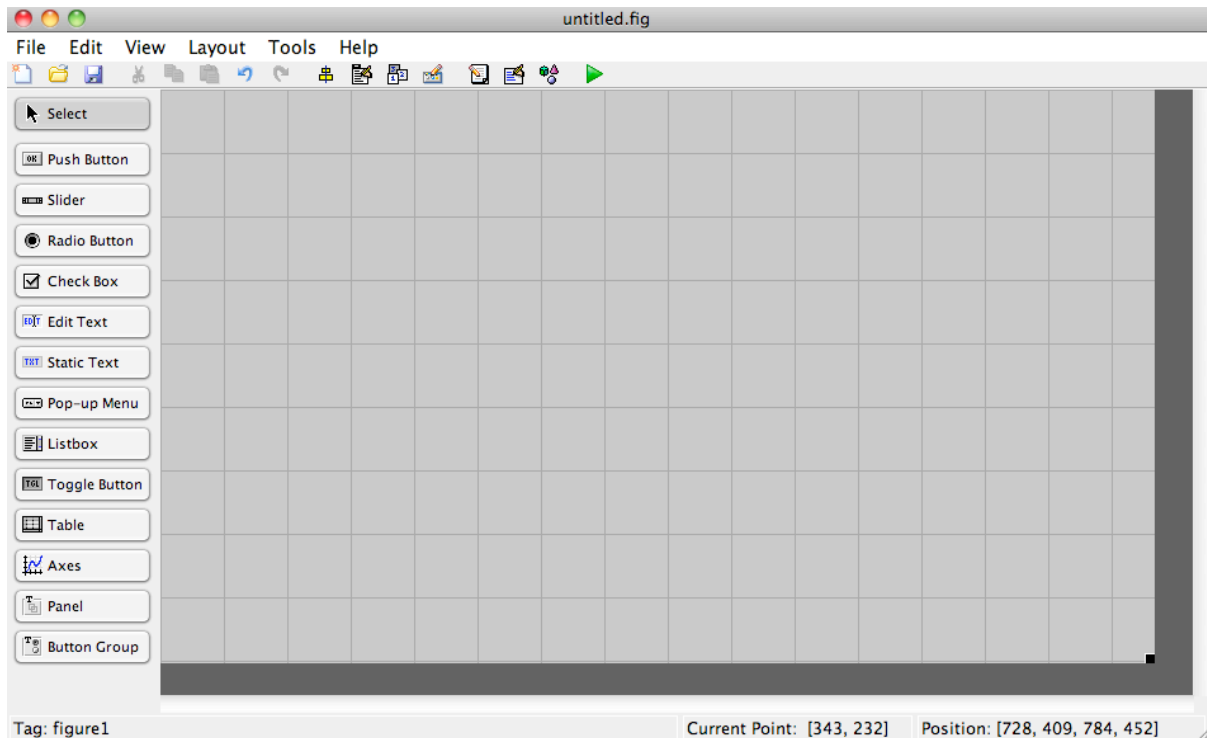


Figura 2.21: Ventana Guide Control Panel

En la que se puede observar en su zona izquierda, todos los uicontrols que han sido descritos previamente, y con los que se realizara la interfaz deseada.

Cada uno de los objetos que compongan el interfaz, podrán ser editados y modificados mediante el Property Inspector (Figura 2.22), donde se podrán observar todas las propiedades que tienen por defecto.

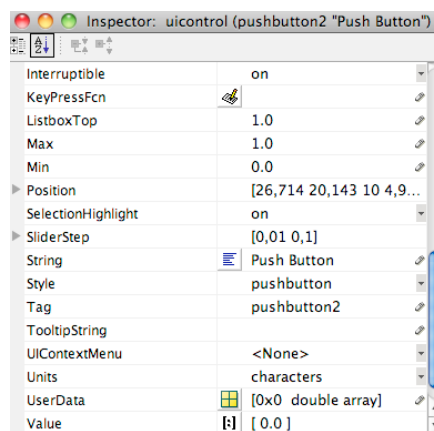


Figura 2.22: Ventana Property Inspector.

Una vez se hallan modificado todas las propiedades de la GUI al gusto del programador, se procederá a guardar los progresos. Y en ese momento se

generaran dos ficheros con el mismo nombre y distintas extensiones, un \*.fig y otro \*.m.

El fichero \*.fig contendrá la figura con el diseño gráfico de la GUI, y el fichero \*.m que contendrá el código necesario para controlar los eventos lanzados por los diferentes controles de la interfaz.

## **2.3. Programas de cálculo de estructuras simplificado: Ed-Tridim.**

### **2.3.1. Introducción**

Este software permite realizar el cálculo de vigas y pórticos hiperestáticos en dos y tres dimensiones, mediante el uso de métodos matriciales [4]. El programa esta organizado en cuatro módulos diferentes:

- Teoría: formado por un conjunto de lecciones teóricas que pueden ser consultadas por el usuario siempre que lo desee; las lecciones no pretenden ser exhaustivas sino tan solo una ayuda para recordar conceptos básicos.
- Biblioteca: permite introducir los datos que definen la estructura a analizar, es decir, la geometría, las propiedades de los materiales, las condiciones de contorno y las cargas que la solicitan; sirve igualmente tanto para crear nuevos problemas como para modificar los existentes.
- Ejemplos: el ordenador realiza paso a paso un problema sin que el usuario tenga que intervenir. El ordenador se convierte en guía, realizando todos los pasos, y deteniéndose en los aspectos mas interesantes de cada etapa.
- Ejercicios: en este modulo, a diferencia del anterior, se hace necesaria la participación activa del usuario. El ordenador asiste al usuario en el dibujo de las leyes de esfuerzos en la estructura.



### 2.3.2. Planteamiento práctico

Como se ha dicho con anterioridad, el módulo Biblioteca es el que permite generar una estructura de cualquier tipo. Al iniciar la sesión e introducirse en el módulo Biblioteca aparece el panel que se muestra en la figura 23:

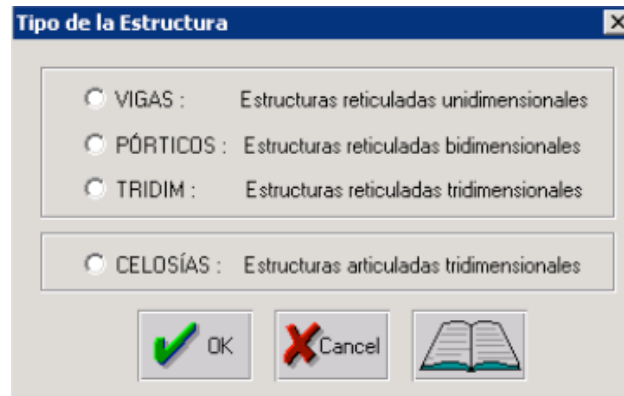


Figura 2.23: Panel para seleccionar el tipo de estructura.

El usuario debe indicar el tipo de estructura que quiere crear:

- VIGAS: permite crear vigas plana unidimensionales.
- PÓRTICOS: permite crear estructuras bidimensionales, ya sean articuladas o reticuladas.
- TRIDIM: permite realizar estructuras tridimensionales reticuladas.
- CELOSÍA: en este último se pueden crear estructuras articuladas tridimensionales.

A continuación hay que dar las cotas espaciales máximas y mínimas de la estructura que se va a crear. Así mismo se puede crear una rejilla de cualquier paso para poder seleccionar con mayor comodidad los puntos de la estructura. Una vez definidas las características generales de la estructura, se procede a la definición concreta de cada uno de los componentes de ésta.

La barra de botones que aparece es la siguiente:



Figura 2.24: Barra de botones del módulo Biblioteca.

A continuación se van a enumerar uno a uno los pasos necesarios para definir la estructura correctamente. Estos sirven igualmente para cualquier tipo de estructura:

1. Generación de los nodos: en primer lugar es necesario identificar cuales son los nodos que componen la estructura; existen nodos de dos tipos: rígidos o articulados, por lo tanto además de la localización habrá que tener cuidado al seleccionar el tipo de nodo adecuado. La selección del tipo de nodo se hace presionado el botón de nodo articulado, en caso de ser un nodo rígido, se presiona primero el nodo articulado y posteriormente se activa el botón de nodo rígido.
2. Generación de secciones y materiales: a continuación se tienen que identificar las propiedades del material y, o bien su geometría (en caso de ser una sección rectangular), o bien su momento de inercia. Una vez se ha definido un material, en la parte superior izquierda del cuadro de dibujo aparece un icono con la forma del tipo de sección elegida para ese material; en el caso de generar varios materiales, habrá varios iconos, cada uno correspondiente a un material. Pinchando con el ratón sobre dicho icono, se pueden ver las propiedades. Lógicamente el icono puede tener varias formas según se haya definido una sección arbitraria, una sección cuadrada, o una sección que va a trabajar solo en axil.
3. Generación de barras: una vez definidos el material o los materiales, se tienen que identificar las barras. Esta operación se realiza pinchando  
  
en los nodos que serán origen y final de la barra en cuestión. Cuando se selecciona un nodo, éste cambia de color; de esta manera se puede saber si se está realizando bien la operación.
4. Generación de los apoyos: este es el último apartado en la generación de la geometría de la estructura. En primer lugar se activa el icono de apoyos, después se selecciona el nodo en el que se va a colocar dicho apoyo y a continuación, aparecerá una ventana en la cual se tiene que indicar el tipo de apoyo que se quiere imponer.

Una vez concluidos todos estos pasos, conviene salvar la estructura; esta operación se realiza como en cualquier programa que trabaja bajo Windows. En las figuras 26 y 27 se muestran, respectivamente, los cuadros de diálogo que aparecen al definir las secciones y al definir los apoyos.

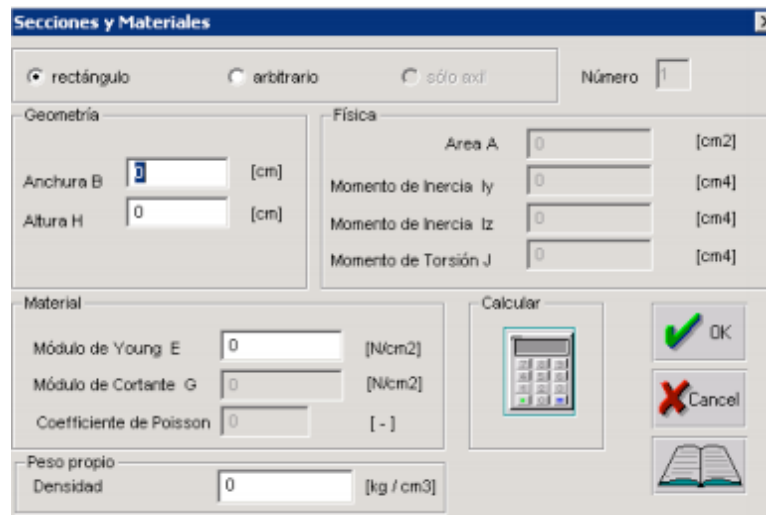


Figura 2.25: Menú donde se elige el tipo de sección.

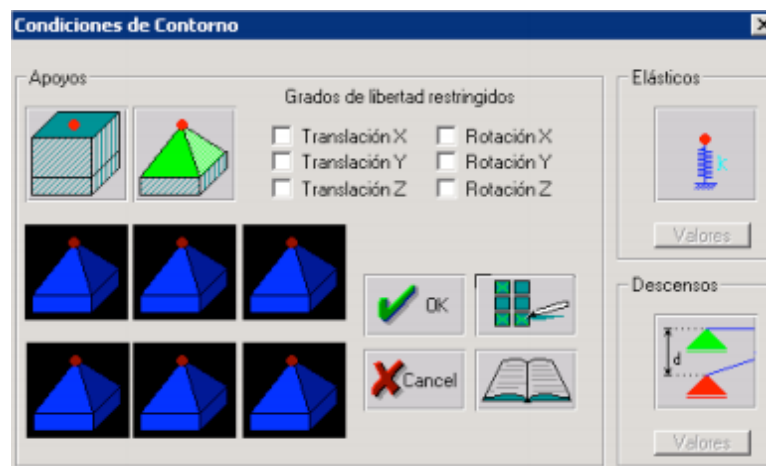


Figura 2.26: Panel para la selección del tipo de apoyos.

### 2.3.3. Resultados

Como se ha explicado anteriormente, los resultados se pueden visualizar utilizando el módulo de Ejercicios o el de Ejemplos. Ambos son muy similares, y la única diferencia estriba en que en el de Ejercicios se requiere de la participación del usuario, mientras que en el de Ejemplos no. En ambos casos el programa permite elegir como se va a resolver la estructura: con solución final o con solución paso a paso. En esta última modalidad, se van visualizando los cálculos que realiza el ordenador internamente.

Concretamente este software utiliza un método matricial, y por lo tanto nos muestra como se van formando las matrices de cada una de las partes de la estructura. A continuación se va a ver someramente como son cada uno de estos módulos.

### 2.3.3.1. Módulo ejemplos

Como se ha dicho con anterioridad, en este módulo, una vez solucionada la estructura, se pueden ver todos los resultados, con tan sólo presionar los botones de la barra de herramientas. Para visualizar las leyes de esfuerzos o deformadas, únicamente hay que presionar el botón correspondiente; en cambio, para ver las reacciones en un apoyo o los desplazamientos de un nudo, hay que mantener presionado el botón correspondiente y pinchar con el ratón en el lugar (nodo o apoyo) donde se quiera conocer un determinado resultado. En la figura 5 se puede ver la barra de botones de los módulos en los que se soluciona la estructura.

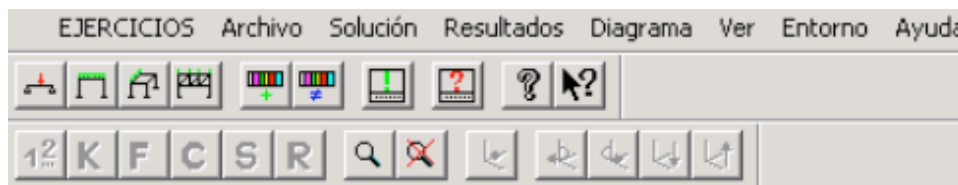


Figura 2.27: Barra de botones de los módulos Ejemplos y Ejercicios.

### 2.3.3.2. Módulo Ejercicios

Este módulo es un poco diferente al anterior. Una vez alcanzada la solución final, ya sea paso a paso o directamente, se pueden consultar los iconos de la barra de tareas que dan los movimientos en los nodos y las reacciones, mientras que no así los de las leyes de esfuerzos.

Para obtener estas leyes primero hay que pinchar sobre una de las barras y a continuación presionar uno de los botones que las dan. En ese momento en la pantalla tan solo aparecerá la barra en cuestión, y utilizando los iconos con las echas, se va dibujando el diagrama de esfuerzos correspondiente. Para volver a la pantalla de la estructura una vez que se ha dibujado el diagrama, hay que pulsar el icono de salir.

La figura 2.28 muestra la pantalla que aparece al dibujar interactivamente las leyes de esfuerzos, particularmente el caso de los axiles:

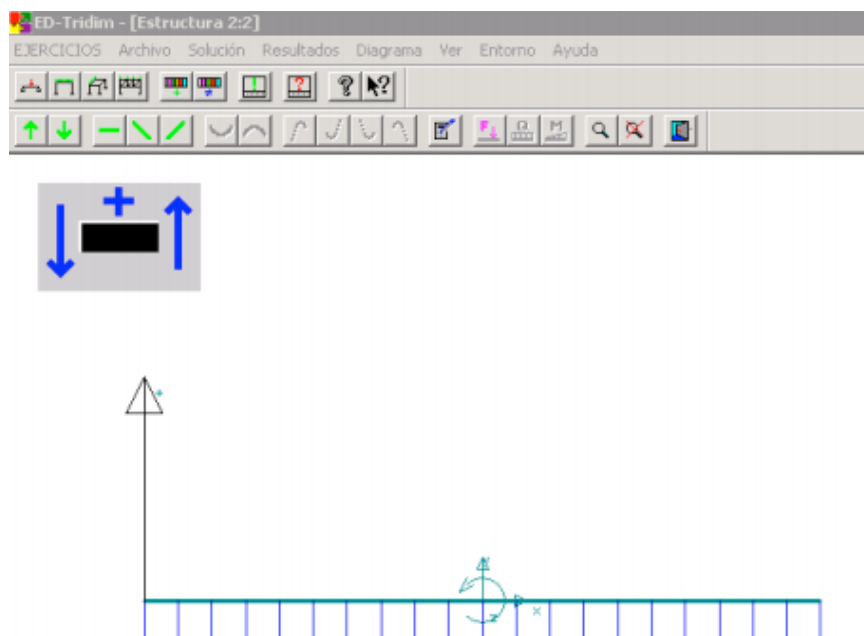


Figura 2.28: Ventana con el dibujo de las leyes de esfuerzos.

## 2.4. CypeCAD

### 2.4.1. Introducción

CypeCAD es un programa de cálculo de estructuras desarrollado por CYPE ingenieros S.A.. La primera versión del programa data de 1983, bajo entorno MS-DOS y en modo texto. La primera versión para Windows aparece en 1997. Existen continuas versiones de mejora y actualización del programa, cuyo número de versión coincide con el año de comercialización del producto, más un número (o número y letra) para indicar revisiones menores. CypeCAD es uno de los programas de cálculo más extendidos en arquitectura y obra civil en España, con aproximadamente 48.000 profesionales registrados.

CypeCAD ha sido concebido para realizar el cálculo y dimensionamiento de estructuras de hormigón armado y metálicas, sometidas a acciones horizontales y verticales, para viviendas, edificios y proyectos de obra civil <sup>[3]</sup>.

### 2.4.2. Características diferenciales de CypeCAD

- Introducción automática de obras: con el módulo Introducción automática de obras: DXF, DWG y modelos CAD/BIM de CYPECAD se dispone de dos opciones que permiten generar automáticamente la estructura mediante la importación de ficheros en formato IFC generados por los principales programas CAD/BIM (Allplan®, Archicad®, Revit® Architecture); o mediante la interpretación de ficheros en formato DXF y DWG.
- Gran potencia de cálculo: se obtendrá la inversión de la matriz de rigidez global de las estructuras en el menor tiempo posible gracias a los métodos de condensación de subestructuras y a la resolución del sistema de ecuaciones por métodos frontales. Sin límite de nudos y barras, en la práctica habitual.
- Cálculo sísmico: análisis modal espectral completo que resuelve cada modo como una hipótesis y realiza la expansión modal y la combinación modal para la obtención de esfuerzos; todo ello sin necesidad de producir fuerzas estáticas equivalentes sísmicas, que es el método simplificado que utilizan otros.
- Cálculo personalizado: se dispone de una gran cantidad de opciones de cálculo y armado que permite hacer las consideraciones que se estimen más adecuadas. Además, para cada elemento estructural y cada posición de armaduras se puede definir tablas de armaduras personalizadas.
- Se puede personalizar los planos según necesidades, ya que el programa permite configurar todas las capas y elementos del dibujo, y generarlos en DXF, DWG, impresora y plotter.

- Versatilidad de la cimentación: las zapatas y encepados permiten disponer sobre cada elemento de cimentación varios soportes, ya sea del mismo tipo o combinando pilares o columnas, pantallas y tramos de muros.
- Con el módulo Cálculo avanzado de cimentaciones superficiales es posible: diseñar cimentaciones con zapatas o encepados con encuentros especiales (cruces entre vigas de atado y centradoras), establecer límites poligonales para zapatas e introducir cargas lineales, puntuales y superficiales sobre zapatas, encepados, vigas de atado y vigas centradoras.
- Potente editor de armaduras: en todos los elementos de dimensionamiento se puede editar y modificar su geometría y armaduras, con múltiples herramientas que le facilitarán la tarea.
- Listados detallados de las comprobaciones de estados límite últimos (E.L.U) de perfiles de acero laminado, armado y conformado, de aluminio, y de madera. Estos listados contienen todas las comprobaciones realizadas por el programa para dimensionar los perfiles y constituyen un importante documento con el que el usuario puede verificar y optimizar el dimensionamiento de los perfiles.
- Planos completos: son planos de construcción de las estructuras, muy completos, con la posibilidad de componerlos, incluir detalles constructivos, DXF, DWG, cajetines, tablas de medición, etc., y que, de este modo, proporcionan los planos más precisos y detallados para ejecutar la obra.
- Generación y exportación de la medición y presupuesto de la estructura calculada en CypeCAD.
- Dimensionamiento de uniones soldadas y atornilladas de perfiles laminados y armados en doble T con los módulos Uniones I, Uniones II, Uniones III y Uniones IV y de perfiles tubulares con Uniones V.
  
- El programa Memorias CTE de CYPE importa datos de la obra calculada en CYPECAD para la elaboración de la Memoria del proyecto básico y de ejecución de un proyecto.
- Exportación a TEKLA® Structures y a TecnoMETAL ® 4D y al formato CIS/2: el programa dispone de módulos adicionales, comunes a CYPECAD y Nuevo Metal 3D, que permiten exportar a TEKLA Structures, a TecnoMETAL 4D y al formato CIS/2 las estructuras 3D integradas.
- Más de 1.100 detalles componen la extensa biblioteca de detalles constructivos metálicos, de hormigón, mixtos y de forjados inclinados disponibles para incluir en los planos generados por el programa. También puede adquirir esta biblioteca editada en dos volúmenes que incluye los detalles en formato DXF y DWG.

# 3. Descripción del programa



## **3.1. Introducción**

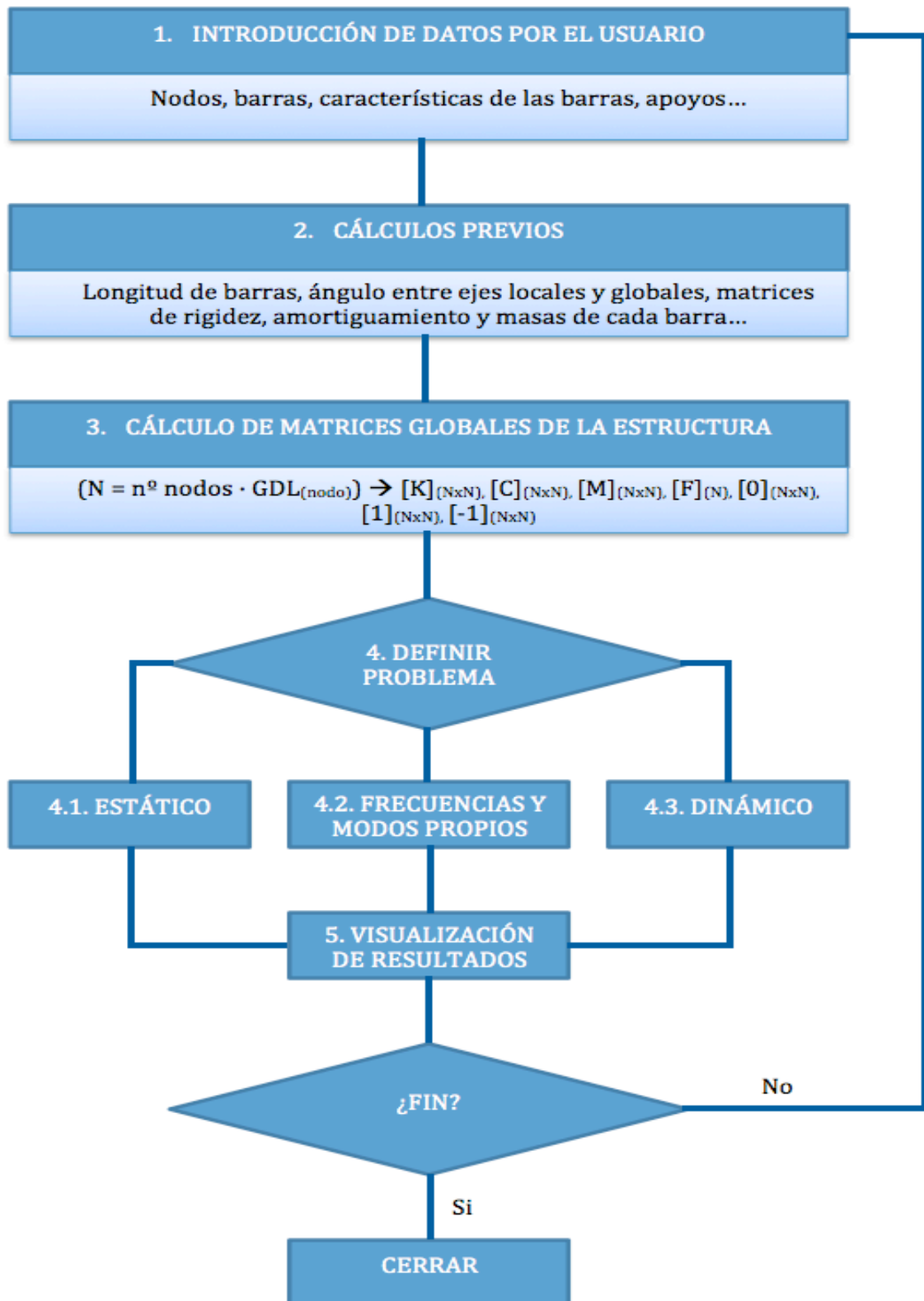
Para la descripción del programa se partirá de un organigrama, que se puede observar en el apartado 3.2., para explicar de forma esquemática los pasos de introducción de datos (primer paso) y posteriores cálculos previos (segundo y tercer paso), para realizar los cálculos de la estructura.

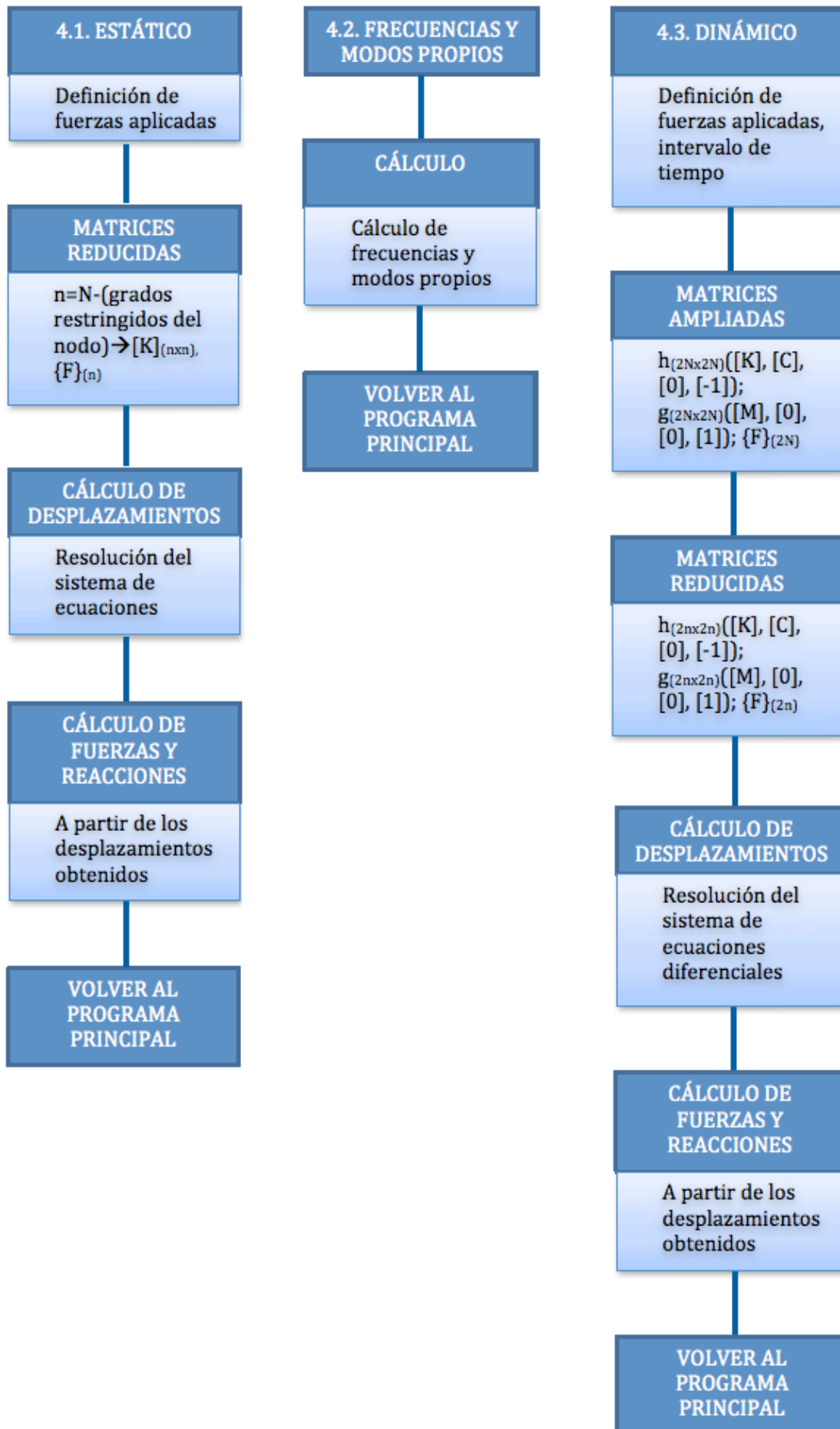
A partir del cuarto paso, el usuario deberá especificar que tipo de problema quiere resolver, cuyo procedimiento vendrá explicado en un organigrama aparte, según sea:

- Caso estático.
- Caso Dinámico.
- Frecuencias y modos propios.

Por último se comentará en detalle cada uno de los pasos del organigrama (apartados 3.3.).

## 3.2. Organigrama





## 3.3. Procedimiento de cálculo

### 3.3.1. Introducción de datos por el usuario

Los datos que se tendrán que introducir para la obtención de desplazamientos y fuerzas en la estructuras serán los estrictamente necesarios, de forma que no exista información redundante. Estos son:

- Número de nodos de la estructura.
- Coordenadas de cada nodo.
- Número de barras de la estructura.
- Nodo menor y mayor, de cada una de las barras.
- Número de apoyos de la estructura.
- Grados de libertad de cada apoyo.
- Número de muelles que tiene la estructura.
- Constante de rigidez de cada muelle y su posición en la estructura.
- Número de amortiguadores que tiene la estructura.
- Constante de amortiguamiento de cada amortiguador y su posición en la estructura.
- Tipos de secciones que tiene la estructura, indicando las características de cada una (densidad, módulo elástico, área y momento de inercia).

### 3.3.2. Cálculos previos

A partir de los datos introducidos por el usuario, el programa realizará unos cálculos generando datos de cada una de las barras, necesarios para la obtención de las matrices globales de la estructura:

- Longitud: a partir de las coordenadas del nodo menor y mayor de cada una de las barras y aplicando Pitágoras, el programa calculará las longitudes de las mismas.
- Ángulo entre ejes locales y globales: este ángulo se puede obtener conociendo las coordenadas de cada uno de los nodos de cada barra de la estructura, mediante relaciones trigonométricas.
- Matrices de rigidez: a partir de las características que tenga la sección de cada barra, su longitud, el ángulo entre ejes globales y locales, y, la constante de rigidez de los muelles que tenga la estructura. Se realizará un cálculo de la matriz de rigidez de cada una de las barras en ejes globales.
- Matrices de amortiguamiento: a partir de la constante de amortiguamiento de los amortiguadores que tenga la estructura, se calculará las matrices de amortiguamiento de las mismas.
- Matrices de masas: cada barra repartirá su peso equitativamente entre sus dos nodos, de forma que, se generará una matriz por nodo con la masa que soportará cada uno.

Las dimensiones de estas matrices serán 2x2 en el caso de estructuras articuladas y 3x3 en el caso de estructuras reticuladas.

### 3.3.3. Cálculo de matrices globales de la estructura

Conocidas las matrices de rigidez y amortiguamiento de cada barra, y matriz de masas de cada nodo de la estructura, el programa calculará con estos datos, las matrices globales de la estructura:

- Matriz de rigidez global de la estructura: conocidas las matrices de rigidez de cada barra y los nodos que tiene cada una, el programa ubicará cada una de las submatrices de cada barra, en la posición que le corresponda en la matriz de rigidez global. Ésta será una matriz cuadrada, con tantas celdillas como número de nodos tenga la estructura.

Cada submatriz será de 3x3 en el caso de estructuras reticuladas, y 2x2 en el caso de estructuras articuladas, es decir, que la matriz de rigidez global de la estructura, tendrá unas dimensiones NxN, teniendo en cuenta que:

$$N = (n^{\circ} \text{ de nodos de la estructura}) * (\text{grados de libertad del nodo}) \quad (3.1)$$

- Matriz de amortiguamiento global de la estructura: los conceptos y procedimiento de cálculo son exactamente los mismos que para la matriz de rigidez global de la estructura, con una diferencia, y es que, en este caso, la matriz de amortiguamiento global de la estructura, se calculará a partir de las matrices de amortiguamiento de cada una de las barras.
- Matriz de masas global de la estructura: consiste en una matriz en la que cada posición de la diagonal principal esta ocupada por cada una de las matrices de masas de cada nodo, siendo el resto de los valores ceros.

Además, se generarán otras tres matrices numéricas:

- Matriz de ceros: todos los valores de la matriz serán cero.
- Matriz de unos: todos los valores de la matriz serán cero menos los de la diagonal principal, cuyos valores serán uno.
- Matriz de menos unos: todos los valores de la matriz serán cero menos los de la diagonal principal, cuyos valores serán menos uno.

Tendrán la misma dimensión que las matrices de rigidez, amortiguamiento y masas. Solo serán necesarias en la resolución de estructuras reticuladas, para conformar las matrices ampliadas.

Como se ha explicado previamente en antecedentes, a la hora de resolver una estructura (ya sea articulada o reticulada) sometida a cargas estáticas, se planteará la ecuación matricial (3.2), a partir de la cual se obtendrá un sistema de ecuaciones que habrá que resolver.

$$\{F\}_{(N)} = [K]_{(NxN)} \cdot \{X\}_{(N)} \quad (3.2)$$

Siendo:

$\{F\} \rightarrow$  Vector de fuerzas.

$[K] \rightarrow$  Matriz de rigidez global de la estructura.

$\{X\} \rightarrow$  Vector de desplazamientos.

### 3.3.3.1. Cálculo de matrices ampliadas de la estructura

Cuando se efectúa un análisis dinámico, las acciones sobre la estructura son función del tiempo, lo que hace movilizar unas fuerzas de inercia unidas a las masas y a la aceleración, planteándose la siguiente ecuación en forma matricial:

$$\{F(t)\}_{(N)} = ([K]_{(NxN)} \cdot \{X\}_{(N)}) + ([M]_{(NxN)} \cdot \{\ddot{X}\}_{(N)}) \quad (3.3)$$

Siendo:

$\{F(t)\}_{(N)} \rightarrow$  Vector de fuerzas aplicadas.

$[M]_{(NxN)} \rightarrow$  Matriz de masas global de la estructura.

$\{\ddot{X}\}_{(N)} \rightarrow$  Vector de aceleración.

Y en caso de que a la estructura se le añada un amortiguador, la ecuación matricial que se plantearía sería la siguiente:

$$\{F(t)\}_{(N)} = ([K]_{(NxN)} \cdot \{X\}_{(N)}) + ([M]_{(NxN)} \cdot \{\ddot{X}\}_{(N)}) + ([C]_{(NxN)} \cdot \{\dot{X}\}_{(N)}) \quad (3.4)$$

Siendo:

$[C]_{(NxN)} \rightarrow$  Matriz de amortiguamiento global de la estructura.

$\{\dot{X}\}_{(N)} \rightarrow$  Vector de velocidades.

Se realizará un cambio de variable para convertir la ecuación (3.4) de segundo orden, en una de primer orden de la siguiente forma:

$$\left\{ \begin{array}{l} \{F(t)\}_{(N)} = ([K]_{(NxN)} \cdot \{X\}_{(N)}) + ([M]_{(NxN)} \cdot \{\ddot{X}\}_{(N)}) + ([C]_{(NxN)} \cdot \{\dot{X}\}_{(N)}) \\ \{X_1\}_{(N)} = \{X\}_{(N)} \\ \{X_2\}_{(N)} = \{\dot{X}\}_{(N)} \\ \{\dot{X}_2\}_{(N)} = \{\ddot{X}\}_{(N)} = \{\ddot{X}\}_{(N)} \end{array} \right\} \Rightarrow \{F(t)\} = ([K] \cdot \{X_1\}) + ([M] \cdot \{X_2\}) + ([C] \cdot \{\dot{X}_2\})$$

Si se plantea de forma matricial este sistema de ecuaciones y cambios de variables, se obtendrán unas matrices ampliadas que estarán formadas por las matrices globales de la estructura, y las matrices de ceros, unos y menos unos, que permitirán plantear los cambios de variables. Quedando de la siguiente forma:

$$\underbrace{\begin{bmatrix} \{0\}_{(N)} \\ \{F(t)\}_{(N)} \end{bmatrix}}_{\{F(t)\}_{(2N)}} = \underbrace{\begin{bmatrix} [1]_{(NxN)} & [0]_{(NxN)} \\ [0]_{(NxN)} & [M]_{(NxN)} \end{bmatrix}}_{[g]_{(2Nx2N)}} \cdot \underbrace{\begin{bmatrix} \dot{\{X_1\}}_{(N)} \\ \dot{\{X_2\}}_{(N)} \end{bmatrix}}_{\dot{\{X\}}_{(2N)}} + \underbrace{\begin{bmatrix} [0]_{(NxN)} & [-1]_{(NxN)} \\ [K]_{(NxN)} & [C]_{(NxN)} \end{bmatrix}}_{[h]_{(2Nx2N)}} \cdot \underbrace{\begin{bmatrix} \{X_1\}_{(N)} \\ \{X_2\}_{(N)} \end{bmatrix}}_{\{X\}_{(2N)}}$$

Por tanto, el programa se encargará de generar, a partir de las matrices globales de la estructura calculadas previamente, las matrices ampliadas “[g]<sub>(NxN)</sub>” y “[h]<sub>(NxN)</sub>”.

Con este planteamiento genérico se podrá resolver estructuras de “n” grados de libertad.

### 3.3.4. Definición del tipo de carga aplicada a la estructura y obtención de valores y modos propios de la estructura.

Llegados a este punto, lo único que quedaría por definir sería el tipo de cargas que se tendrían que aplicar a la estructura:

- Cargas estáticas.
- Cargas dinámicas.

Por otro lado, con los datos aportados y los cálculos previos realizados por el programa, se podrían obtener los valores propios y modos propios de la estructura que se explicará a continuación.

#### 3.3.4.1. Cargas estáticas

Será necesario que el programa genere un vector de fuerzas  $\{F\}_{(N)}$ , tal que, el primer valor del vector será un vector que se corresponderá con el nodo uno, el segundo valor con el nodo dos y así sucesivamente. Así mismo, el vector de cada nodo estará formado por tres o dos valores dependiendo del tipo de estructura (dos en articuladas y tres en reticuladas). El primer valor del vector de un nodo indicará la magnitud de la fuerza en el eje “x”, el segundo será la magnitud de la fuerza en el eje “y” y en el caso de estructuras reticuladas, el tercero será la magnitud del momento aplicado en el eje “z”.

Para ello el usuario deberá indicar en que nodos y con que magnitud, dirección y sentido se va a aplicar la fuerza en cada uno de ellos. Y una vez introducidos, el programa se encargará de ubicar estos valores en el vector  $\{F\}_{(N)}$  y dar valor

cero al resto de valores del vector. De forma que el vector de fuerzas generado para una barra de una estructura reticulada aplicándose una fuerza en el nodo uno, en dirección “y” y en sentido contrario a los ejes de coordenadas será el siguiente:

$$\{F\}_{(N)}^t = \begin{pmatrix} 0 & -F & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.7)$$

### 3.3.4.2. Cargas dinámicas

En el caso de someter a la estructura a cargas dinámicas, además de indicar los mismos datos que en cargas estáticas, se tendrá que introducir la frecuencia angular ( $\omega$ ), el desfase ( $\varphi$ ) y, teniendo en cuenta que la carga dinámica depende del tiempo, será necesario introducir el intervalo de tiempo en el que se quiere estudiar la respuesta de la estructura sometida a dichas cargas.

De forma que para el ejemplo planteado en el apartado anterior, se obtendría el siguiente vector de fuerzas:

$$\{F(t)\}_{(N)}^t = \begin{pmatrix} 0 & -F \cdot \text{sen}\{(\omega \cdot t) + \varphi\} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.8)$$

### 3.3.4.3. Valores y modos propios

Cuando un cuerpo capaz de vibrar es sometido a la acción de una fuerza periódica, cuyo periodo de vibración coincide con el periodo de vibración característico de dicho cuerpo, se produce la resonancia. Por lo que, una fuerza relativamente pequeña aplicada en forma repetida, hace que la amplitud de un sistema oscilante se haga muy grande.

Con las fórmulas:

$$|[K] \cdot [M]^{-1} - \omega \cdot I| = 0 \quad (3.9)$$

$$([K] - \omega^2 \cdot [M]) \cdot x = 0 \quad (3.10)$$

Se pueden obtener, en primer lugar a partir de la fórmula (3.9) los valores propios, que definen las frecuencias y periodos naturales de la estructura. Y una vez obtenidos éstos, la fórmula (3.10) proporcionará la forma modal correspondiente a cada valor propio, que será un modo propio para cada valor, que representara las amplitudes de la deformada de cada modo de vibración. Hay tantos modos de vibración como grados de libertad de la estructura considerada.



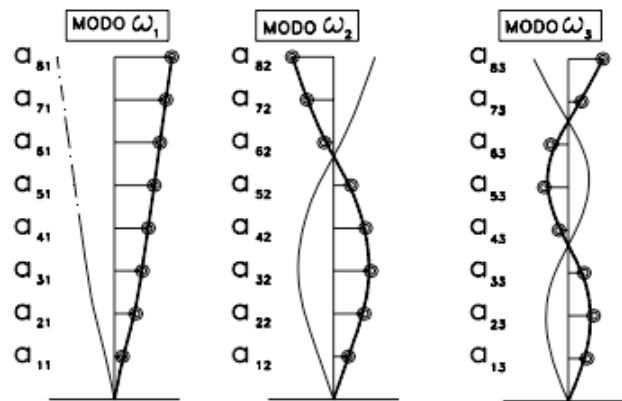


Figura 3.1: Primeros modos de vibración de un oscilador simple.

### 3.3.5. Obtención de las matrices reducidas

Como se comenta en el apartado 2.1.2.2.7 de Antecedentes, para la obtención de desplazamientos, fuerzas y reacciones, habrá que plantear en primer lugar, unas ecuaciones de las que se obtendrán los desplazamientos de la estructura, y posteriormente, conocidos los desplazamientos, se obtendrán las fuerzas y reacciones de la estructura.

Se partirá de una ecuación matricial en la que se contará con todos los grados de libertad de los nodos de la estructura (libres y restringidos). Para la obtención de desplazamientos, será necesario eliminar todas las filas y columnas en las que coincida que el grado de un nodo este restringido, es decir, que su desplazamiento sea nulo. Por lo tanto, habrá que reducir en el caso de estructuras sometidas a cargas estáticas, la matriz de rigidez global de la estructura ( $[K]_{(N \times N)}$ ) y su vector de fuerzas ( $\{F\}_{(N)}$ ) o, en el caso de estructuras sometidas a cargas dinámicas, las matrices ampliadas de la estructura ( $[h]_{(N \times N)}$ ) y  $[g]_{(N \times N)}$  y su vector de fuerzas ( $\{F(t)\}_{(N)}$ ). De forma que la dimensión de las matrices y vectores reducidos será:

$$n = N - (\text{grados restringidos de cada nodo}) \quad (3.11)$$

El programa será capaz, a partir de los datos introducidos por el usuario en los que indica el número de apoyos y sus restricciones, calcular las matrices reducidas de la matriz de rigidez global ( $[k]_{(n \times n)}$ ), su vector de fuerzas ( $\{f\}_{(n)}$ ), las matrices ampliadas ( $[H]_{(n \times n)}$  y  $[G]_{(n \times n)}$ ) y el vector de fuerzas ( $\{f(t)\}_{(n)}$ ) de la estructura.

### 3.3.6. Obtención de desplazamientos

En el caso de estructuras sometidas a cargas estáticas, el programa tendrá planteada una ecuación en la que se despeja el vector  $\{x\}_{(n)}$ , y se obtienen los resultados de los desplazamientos:

$$\{x\}_{(n)} = [k]_{(n \times n)}^{-1} \cdot \{f\}_{(n)} \quad (3.12)$$

Para estructuras sometidas a cargas dinámicas, se obtendrán los desplazamientos y velocidades de los nodos, a partir de la siguiente fórmula:

$$\dot{\{x\}}_{(n)} = [G]_{(n \times n)}^{-1} \cdot (\{f(t)\}_{(n)} - [H]_{(n \times n)} \cdot \{x\}_{(n)}) \quad (3.13)$$

### 3.3.7. Obtención de fuerzas y reacciones

Para el cálculo de las reacciones y fuerzas de la estructura, el programa creará un vector desplazamientos  $\{X\}_{(N)}$ , a partir de los desplazamientos obtenidos en el paso previo, y en el que se encontrarán tanto los desplazamientos de los nodos no restringidos, como el de los nodos restringidos (apoyos), cuyo valor sería cero. De forma que, se podrá obtener el vector de fuerzas  $\{F\}_{(N)}$ , que contendrá el valor de las fuerzas y reacciones generadas en cada nodo de la estructura.

Esto será posible para estructuras sometidas a cargas estáticas mediante la ecuación (3.2), y para estructuras sometidas a cargas dinámicas a partir de la ecuación (3.4).

### 3.3.8. Visualización de resultados

Se podrán mostrar los resultados numéricos de los valores y modos propios, desplazamientos, fuerzas y reacciones, de cualquier tipo de estructura sometida a cualquier tipo de carga.

Además, para estructuras sometidas a cargas dinámicas, se podrá mostrar en gráficas, los resultados de los desplazamientos, fuerzas y reacciones de la estructura frente al tiempo. Pudiéndose comparar desplazamientos y fuerzas de distintos nodos de la estructura, e incluso, comparar desplazamientos de nodos de estructuras diferentes.

## 3.4. Interfaz

### 3.4.1. Menú Principal

Para la ejecución del programa se deberá escribir en la ventana de comandos de Matlab lo siguiente:

```
>> Menu_Principal
```

Que es el nombre del fichero que engloba y relaciona los distintos ficheros programados para calcular y obtener los valores de las frecuencias, modos propios, desplazamientos y fuerzas, de las distintas estructuras que el usuario se plantee resolver.



Figura 3.2: Ventana del Menú Principal del programa.

Como se puede observar en la figura 1, en el Menú Principal hay una serie de botones con distintas funcionalidades y programaciones:

- Botones de definición de estructuras articuladas y reticuladas: al clicar sobre estos botones, el usuario podrá definir la estructura que quiera analizar.
- Botones de fuerzas estáticas o dinámicas: al clicar sobre estos botones, el usuario podrá definir la magnitud de la fuerza a la que quiera someter a la estructura.
- Botones de frecuencias y modos propios: al clicar sobre estos botones, el usuario podrá conocer las frecuencias y modos propios de la estructura que ha definido.

- Botones de calculo de desplazamientos y fuerzas: al clicar sobre estos botones, el usuario podrá conocer los desplazamientos y reacciones generadas en la estructura bajo las condiciones impuestas por el usuario.

### 3.4.2. Definición de datos de la estructura

En el momento en que el usuario clique con el ratón sobre el botón *Definición Estructura Reticulada (E.R.)* o *Definición Estructura Articulada (E.R.)*, inmediatamente saltará una ventana como la que se muestra en la figura 2, en la que el usuario, en 11 pasos, deberá introducir toda la información que se le solicite.

The screenshot shows a software window titled "Definicion\_Estructura\_Reticulada" with a blue background. It contains 12 numbered steps for defining a truss structure, each with a corresponding input field or table.

1. Número de nodos de la estructura: Input field.
2. Coordenadas de cada nodo: Table with columns: Nodo, Coordenada X, Coordenada Y.
3. Número de barras de la estructura: Input field.
4. Defina la posición de cada barra: Table with columns: Barra, Nodo menor, Nodo mayor.
5. Número de nodos con algun GDL restringidos: Input field.
6. Especifique los grados restringidos: Table with columns: Nodo, Desplazamiento en X, Desplazamiento en Y, Desplazamiento en Z.
7. Número de muelles que hay en la estructura: Input field.
8. Defina la posición de cada muelle: Table with columns: Barra, Constante de rigidez del muelle.
9. (Not explicitly numbered but present): Input field.
10. Número de amortiguadores que hay en la estructura: Input field.
11. Defina la posición de cada amortiguador: Table with columns: Barra, Constante de amortiguamiento del amortiguador.
12. Defina las características de las barras: Table with columns: Barra, Modulo elástico (E), Área, Momento de Inercia (I), Densidad.

A "Cerrar" button is located at the bottom right of the window.

Figura 3.3: Ventana Definición estructura reticulada.

### 3.4.2.1. Definición del número de nodos de la estructura

El clicar sobre el botón 1. *Número de nodos de la estructura*, saltará una ventana (Figura 3.3) llamada *Numero Nodos ER*.

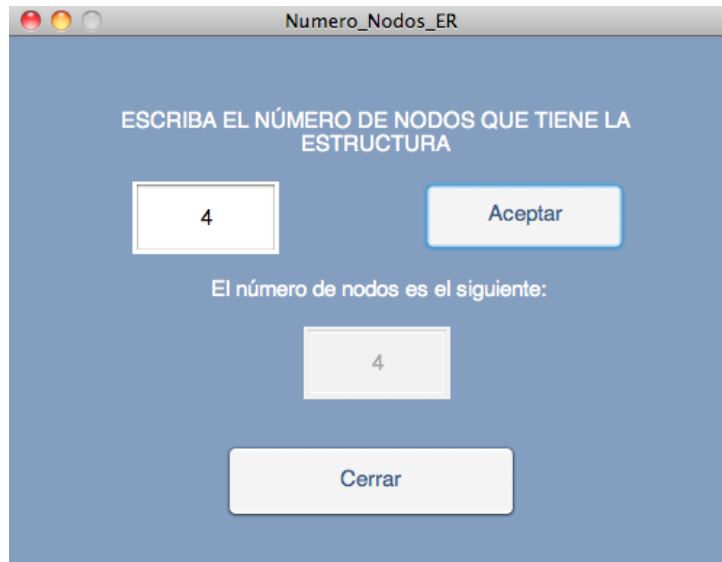


Figura 3.4: Ventana NumNodosr

El usuario introducirá la información en el cuadro de edición de texto editable que aparece a la izquierda del botón *Aceptar* y automáticamente, este número aparecerá en el cuadro de edición de texto no editable posicionado en medio de la ventana en un tono mas grisáceo, que le servirá para confirmar que el programa ha guardado esa información. Posteriormente se pulsará el botón *Cerrar*, desapareciendo la ventana *Numero Nodos ER* y apareciendo en ese momento en la ventana de *Definición Estructura Reticulada*, justo debajo del botón 1. *Número de nodos de la estructura* la información introducida.

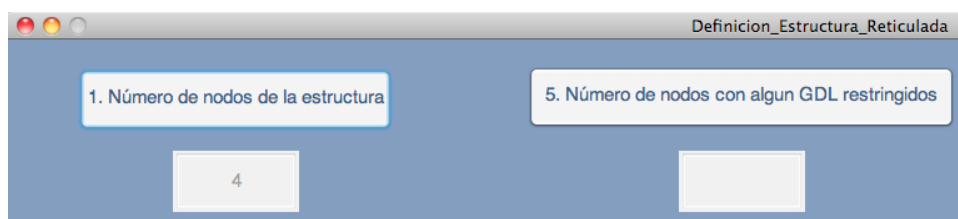


Figura 3.5: Cuadro edición de texto del número de nodos.

### 3.4.2.2. Coordenadas de cada nodo

Conocido el numero de nodos que tiene la estructura, el siguiente botón que habrá que pulsar será: 2. *Coordenadas de cada nodo*. Éste abrirá una ventana como la siguiente:

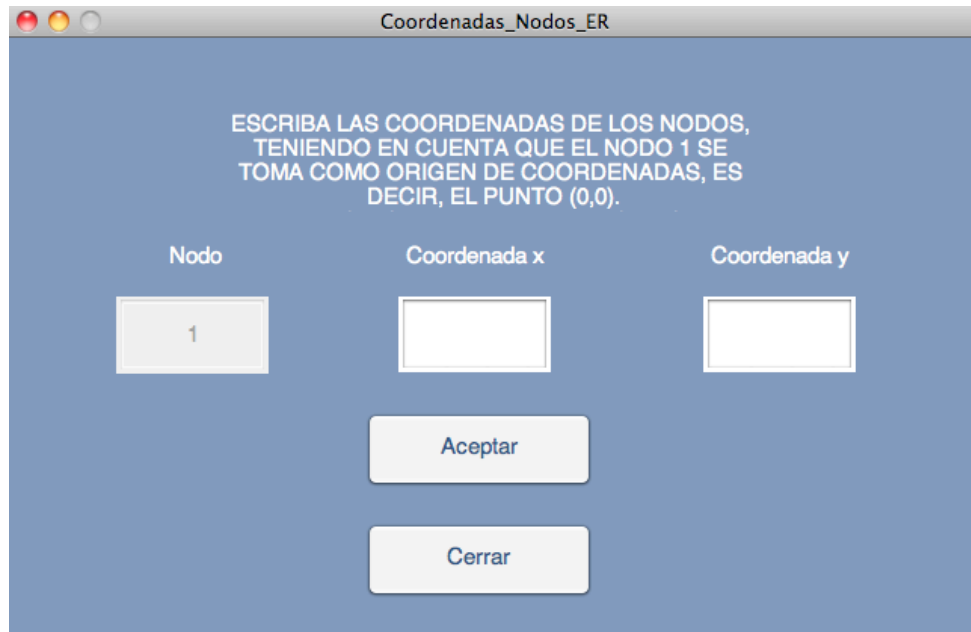


Figura 3.6: Ventana de introducción de coordenadas de cada nodo.

La caja de edición no editable de la izquierda, tiene la función de ir indicando al usuario el número de nodo del que tendría que introducir las coordenadas.

A su vez en las otras dos cajas de edición de texto editables, como indica encima suyo, el usuario deberá de introducir las coordenadas, en las unidades que considere oportuno, siempre que sea consecuente con los demás datos que va a introducir.

Esta ventana, se cerrará y abrirá tantas veces como número de nodos halla indicado el usuario previamente, es decir, si ha indicado que son 4 nodos, el usuario introducirá las coordenadas para el nodo 1, pulsará aceptar y la ventana se cerrará automáticamente, volviéndose a abrir al instante, mostrando en la caja de edición de la izquierda el número dos y así consecutivamente hasta que al completar las coordenadas del nodo 4, se cierre definitivamente.

El botón *Cerrar* está simplemente, por si el usuario ya ha introducido las coordenadas, y sin querer desde la ventana *Definición Estructura Reticulada*, pulsa de nuevo el botón de definición de coordenadas. Se le abrirá de nuevo la ventana, y para no tener que introducir de nuevo las coordenadas, pulsara cerrar y podrá volver a la ventana *Definición Estructura Reticulada*.

Realizado este proceso, el usuario podrá observar en la ventana *Definición Estructura Reticulada* el valor asignado a cada uno de sus nodos en una tabla:

The screenshot shows a software window titled 'Definicion\_Estructura\_Reticulada'. It contains several input fields and a table. Field 1 is 'Número de nodos de la estructura' with a value of 4. Field 2 is 'Coordenadas de cada nodo'. Field 5 is 'Número de nodos con algun GDL restringidos'. Field 6 is 'Especifique los grados restringidos'. A table with 4 rows and 3 columns (Nodo, Coordenada X, Coordenada Y) is highlighted with a red box. The table contains the following data:

Nodo	Coordenada X	Coordenada Y
1	0	0
2	-0.3530	-0.8530
3	0	-0.5000
4	0.3530	-0.8530

Figura 3.7: Tabla con el valor de las coordenadas de cada nodo.

### 3.4.2.3. Número de barras de la estructura

En el tercer paso, el usuario indicará el numero de barras que tiene la estructura. Para ello clicara el botón 3. *Número de barras de la estructura* y saltará una ventana, que funcionará exactamente igual que para la introducción del número de nodos, con el aspecto que se muestra en la dos siguientes imágenes:

The screenshot shows a software window titled 'Numero\_Barras\_ER'. It contains a prompt 'ESCRIBA EL NÚMERO DE BARRAS QUE TIENE LA ESTRUCTURA' and an input field with the value 1. There are two buttons: 'Aceptar' and 'Cerrar'. Below the input field, it says 'El número de barras es el siguiente:' followed by another input field. At the bottom, it says 'Si es correcto pulse 'Cerrar''.

Figura 3.8: Ventana para introducir el número de barras de la estructura.

Definicion\_Estructura\_Reticulada

1. Número de nodos de la estructura

4

2. Coordenadas de cada nodo

Nodo	Coordenada X	Coordenada Y
1	0	0
2	-0.3530	-0.8530
3	0	-0.5000
4	0.3530	-0.8530

3. Número de barras de la estructura

3

5. Número de nodos con algun GDL restringidos

6. Especifique los grados restringidos

Nodo	Desplazamiento en X	Desplazamiento en Y	Desplazamiento

7. Número de muelles que hay en la estructura

Figura 3.9: Indicación del número de barras introducido.

#### 3.4.2.4. Nodos de cada barra

Conocido el número de nodos que tiene la estructura, el siguiente botón que habrá que pulsar será: *4. Defina la posición de cada barra*. Éste abrirá una ventana como la siguiente:

Nodos\_de\_Barras\_ER

INDIQUE LOS NODOS DE CADA BARRA

Barra

1

Nodo menor

|

Nodo mayor

Aceptar

Figura 3.10: Ventana para indicar el nodo de menor y mayor numeración de la barra

El procedimiento es el mismo que el que se emplea para la obtención de las coordenadas. Y una vez especificados los nodos de cada una de las barras se podrá observar los resultados de los datos introducidos en la ventana principal de definición de estructuras reticuladas.



Definicion\_Estructura\_Reticulada

1. Número de nodos de la estructura

4

2. Coordenadas de cada nodo

Nodo	Coordenada X	Coordenada Y
1	0	0
2	-0.3530	-0.8530
3	0	-0.5000
4	0.3530	-0.8530

3. Número de barras de la estructura

3

4. Defina la posición de cada barra

Barra	Nodo menor	Nodo mayor
1	1	3
2	2	3
3	3	4

5. Número de nodos con algún GDL restringidos

6. Especifique los grados restringidos

Nodo	Desplazamiento en X	Desplazamiento en Y	Desplazamiento
------	---------------------	---------------------	----------------

7. Número de muelles que hay en la estructura

8. Defina la posición de cada muelle

Barra	Constante de rigidez del muelle
-------	---------------------------------

Figura 3.11: Representación de los nodos de cada barra en la ventana Definición Estructuras Reticuladas.

### 3.4.2.5. Número de apoyos que tiene la estructura

En el quinto paso, el usuario indicará el número de barras que tiene la estructura. Para ello clicara el botón 5. *Número de nodos con algún GDL restringido* y saltará una ventana, que funcionará exactamente igual que para la introducción del número de nodos, con el aspecto que se muestra en las dos siguientes imágenes:

Numero\_Nodos\_Restringidos\_ER

INDIQUE EL NÚMERO DE APOYOS QUE TIENE LA ESTRUCTURA

Aceptar

El número de apoyos es el siguiente:

Si es correcto pulse 'Cerrar'

Cerrar

Figura 3.12: Ventana para introducir el número de apoyos que tiene la estructura

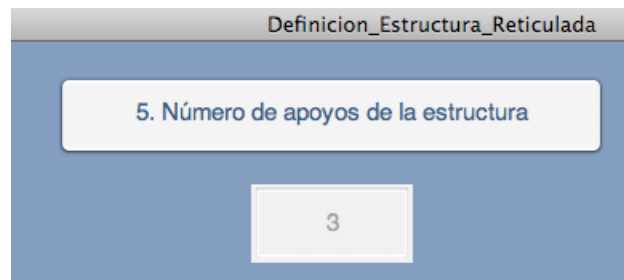


Figura 3.13: Verificación de los datos introducidos en la ventana Definición Estructura Reticulada

### 3.4.2.6. Especificación del tipo de apoyo

Al clicar sobre el botón: 6. *Especifique el tipo de apoyo*, se abrirá una ventana como la siguiente:

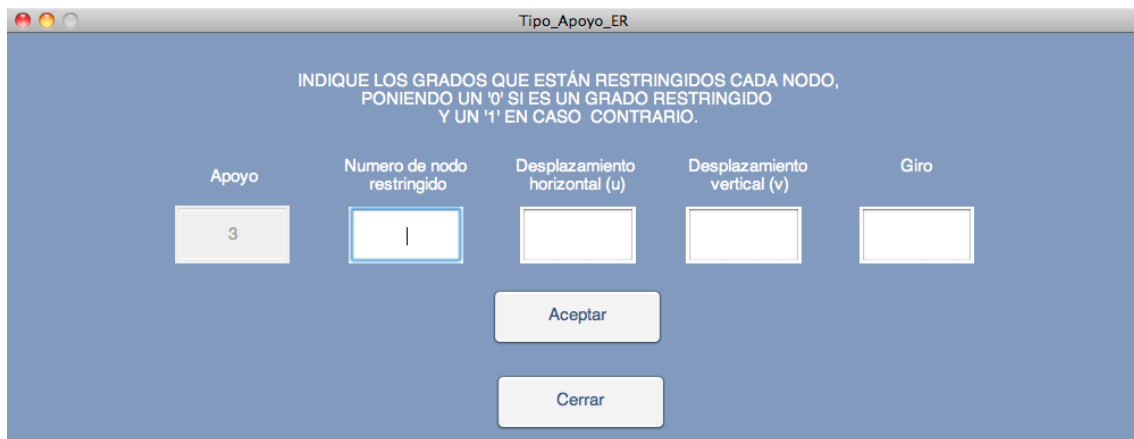


Figura 3.14: Ventana para especificar el tipo de apoyo.

La caja de edición no editable de la izquierda, tiene la función de ir indicando al usuario el número de apoyo que va a definir.

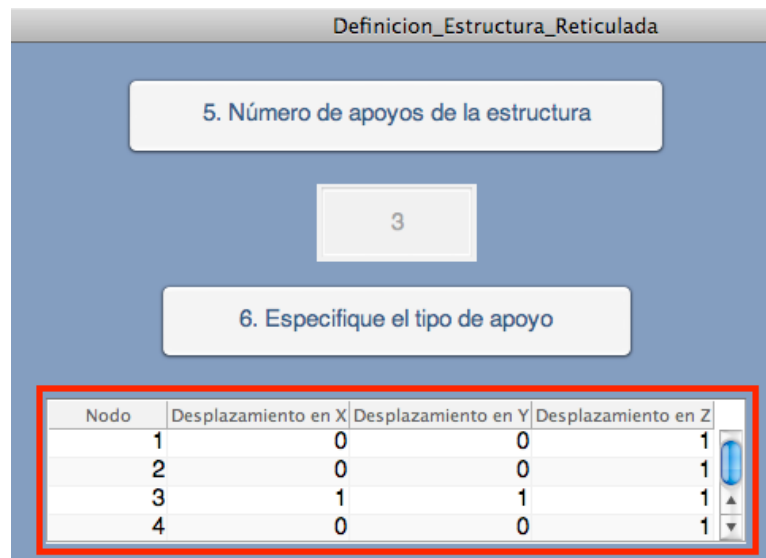
En la segunda caja (empezando por la izquierda), el usuario especificará en que nodo se encuentra el apoyo.

Y por último, en las tres ultimas cajas (si empezamos a contar por la izquierda), el usuario indicará si se produce desplazamiento (o no) horizontal, vertical o giro en el nodo. Para ello habrá que colocar un '0' en caso de que el desplazamiento horizontal, vertical o el giro sean nulos, es decir, tengan dicho grado de libertad restringido. Y con un '1' en caso contrario.

Si se tratara de una estructura articulada, la ventana sería exactamente igual, pero sin giro, por tanto habría cuatro cajas editables, todas menos la de Giro.

Esta ventana, se cerrará y abrirá tantas veces como número de apoyos halla indicado el usuario previamente.

Realizado este proceso, el usuario podrá observar en la ventana *Definición Estructura Reticulada* el valor asignado a cada uno de sus nodos en una tabla:



Nodo	Desplazamiento en X	Desplazamiento en Y	Desplazamiento en Z
1	0	0	1
2	0	0	1
3	1	1	1
4	0	0	1

Figura 3.15: Verificación de los grados de libertad de cada nodo en la ventana Definición Estructura Reticulada

### 3.4.2.7. Número de muelles que tiene la estructura

En el séptimo paso, el usuario indicará el número de muelles que tiene la estructura. Para ello clicará el botón 7. *Número de muelles que hay en la estructura* y saltará una ventana, que funcionará exactamente igual que para la introducción del número de nodos, con el aspecto que se muestra en la siguiente imagen:

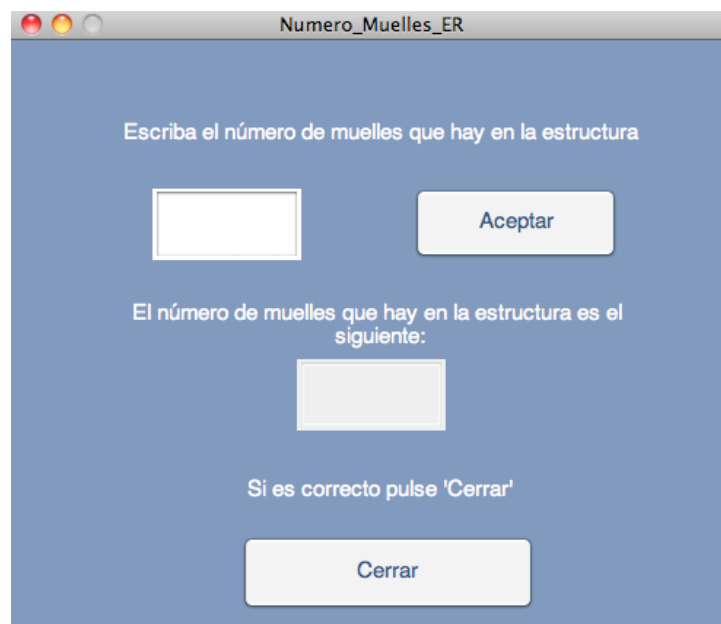


Figura 3.16: Ventana para introducir el número de muelles que tiene la estructura.

### 3.4.2.8. Definición de la constante de rigidez de los muelles de la estructura

Al clicar sobre el botón: 8. *Defina la posición de cada muelle*, se abrirá una ventana como la siguiente:

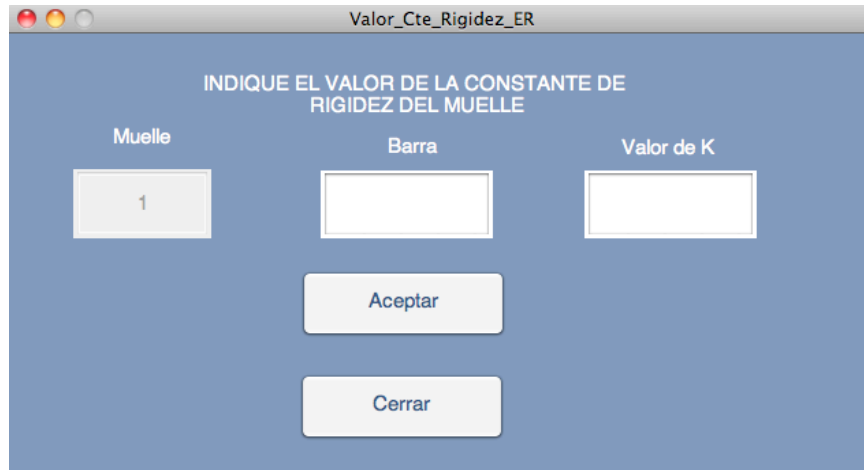


Figura 3.17: Ventana en la que se indica que elemento de la estructura es un muelle y el valor de su constante de rigidez.

La caja de edición de datos no editable de la izquierda, tiene la función de ir indicando al usuario el número de muelle que va a definir.

En la caja de edición de datos que aparece en la zona central de la ventana, el usuario indicará qué barra se considera un muelle. Y en la que pone *Valor de K*, introducirá el valor de la constante de rigidez de dicho muelle.

El botón *Cerrar* se empleará, por si el usuario ya ha introducido los datos, y quiere salir directamente a la ventana *Definición Estructura Reticulada*.

Realizado este proceso, el usuario podrá observar en la ventana *Definición Estructura Reticulada* qué elementos son muelles, comprobando que la barra se corresponde con el valor de la constante de rigidez mostrado en la tabla:

Definicion\_Estructura\_Reticulada

5. Número de apoyos de la estructura

3

6. Especifique el tipo de apoyo

Nodo	Desplazamiento en X	Desplazamiento en Y	Desplazamiento en Z
1	0	0	0
2	0	0	0
3	1	1	1
4	0	0	0

7. Número de muelles que hay en la estructura

1

8. Defina la posición de cada muelle

Barra	Constante de rigidez del muelle
1	1000000
2	0
3	0

Figura 3.18: Tabla en la que se muestra que la barra 1 es un muelle con valor  $K=1000000\text{N/m}$

### 3.4.2.9. Número de amortiguadores que tiene la estructura

En el noveno paso, el usuario indicará el número de amortiguadores que tiene la estructura. Para ello clicará el botón 9. *Número de amortiguadores que hay en la estructura* y saltará una ventana, que funcionará exactamente igual que para la introducción del número de nodos, con el aspecto que se muestra en la siguiente imágenes:

Numero\_Amortiguadores\_ER

ESCRIBA EL NÚMERO DE AMORTIGUADORES QUE HAY EN LA ESTRUCTURA

1

Aceptar

El número de amortiguadores que hay en la estructura es el siguiente:

1

Si es correcto pulse 'Cerrar'

Cerrar

Figura 3.19: Ventana para introducir el número de amortiguadores que tiene la estructura.

### 3.4.2.10. Definición de la constante de amortiguamiento de los amortiguadores de la estructura

Al clicar sobre el botón: 10. Defina la posición de cada amortiguador, se abrirá una ventana como la siguiente:

Figura 3.20: Ventana en la que se indica que elemento de la estructura es un amortiguador y el valor de su constante de amortiguamiento.

La caja de edición de datos no editable de la izquierda, tiene la función de ir indicando al usuario el número de amortiguador que va a definir.

En la caja de edición de datos que aparece en la zona central de la ventana, el usuario indicará qué barra se considera un amortiguador. Y en la que pone *Valor de C*, introducirá el valor de la constante de amortiguamiento de dicho amortiguador.

El botón *Cerrar* se empleará, por si el usuario ya ha introducido los datos, y quiere salir directamente a la ventana *Definición Estructura Reticulada*.

Realizado este proceso, se podrá observar en la ventana *Definición Estructura Reticulada* qué elementos son amortiguadores, comprobando que la barra se corresponde con el valor de la constante de amortiguamiento mostrado en la tabla:

Barra	Constante de amortiguamiento del amortiguador
1	1000
2	0
3	0

Figura 3.21: Tabla en la que se muestra que la barra 1 es un amortiguador con valor  $C=1000 \text{ N}\cdot\text{s/m}$

### 3.4.2.11. Definición de la sección de las barras de la estructura

Al clicar sobre el botón: *11. Defina las características de las barras*, se abrirá la siguiente ventana, con distintos apartados:

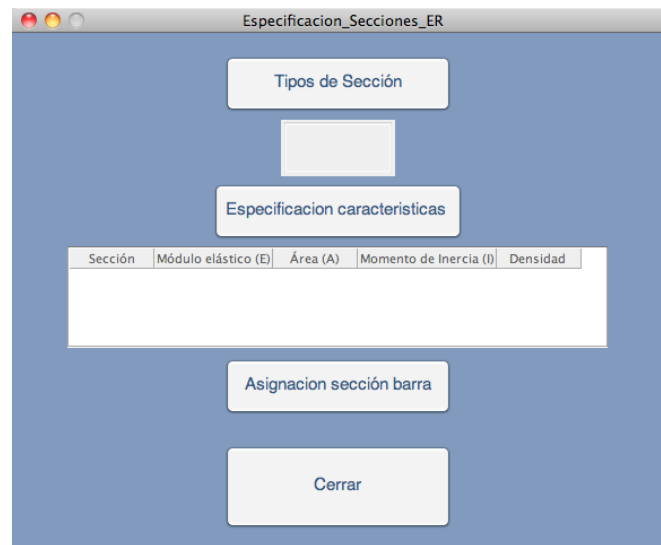


Figura 3.22: Ventana para especificar la sección de cada una de las barras.

#### 3.4.2.11.1. Botón Tipos de Sección

Al clicar sobre este botón se abrirá la siguiente ventana:

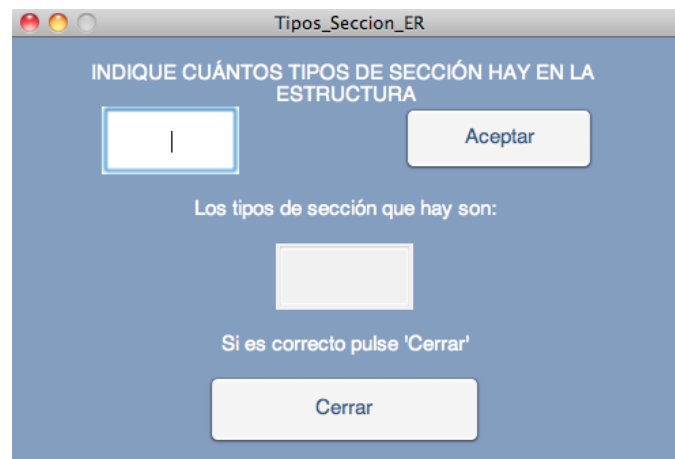


Figura 3.23: Ventana para introducir el número de distintas secciones que hay en la estructura.

El usuario introducirá el dato solicitado, clicará sobre el botón *Aceptar* y posteriormente *Cerrar*.

Mediante este botón se podrá especificar cuantos tipos de secciones distintas hay en las barras de la estructura.

### 3.4.2.11.2. Botón Especificación Características

Al clicar sobre este botón, aparecerá la siguiente ventana:

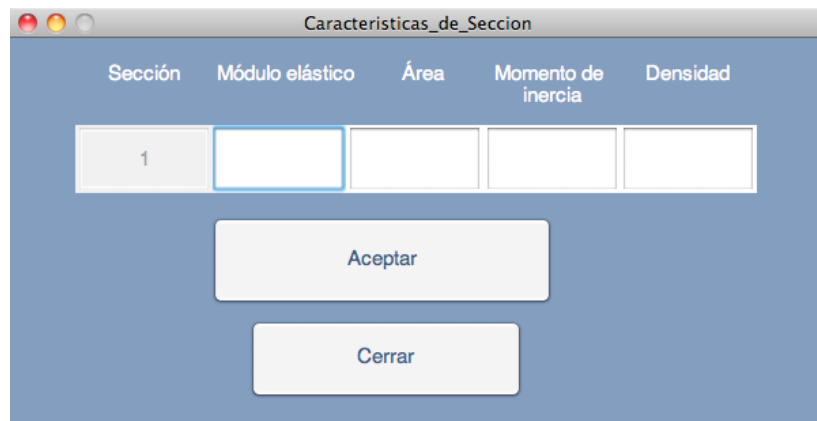


Figura 3.24: Ventana para especificar las características de la sección

La caja de edición de texto no editable de la izquierda, irá indicando la sección que se esta definiendo.

Se deberá especificar el módulo elástico, área, densidad y momento de inercia de la sección. Para estructuras articuladas no será necesario indicar el valor de momento de inercia.

La ventana se cerrará y aparecerá tantas veces como secciones halla indicado el usuario previamente.

El botón Cerrar se usará en caso de que no se quiera introducir datos.

### 3.4.2.11.3. Botón Asignación sección barra

Al clicar sobre este botón, aparecerá la siguiente ventana:

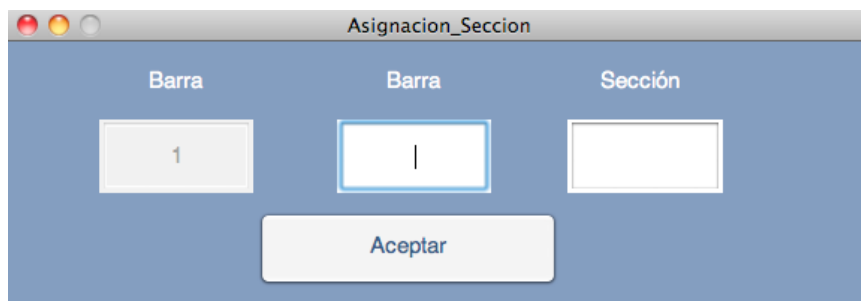


Figura 3.25: Ventana para asignar sección a cada barra.

La caja de edición de texto no editable de la izquierda, tiene la función de ir recordando al usuario el número de barra a la que tiene que asignar una sección.



En caso de que solo exista una única sección, esa ventana no se abrirá y se asignará automáticamente esa sección a todas las barras.

Como último paso se cerrará la ventana Especificacion\_Secciones\_ER y se podrá observar las características de cada barra en la tabla que se encuentra debajo del botón 11.

The screenshot shows a software window titled 'Definicion\_Estructura\_Reticulada'. It contains several input fields and tables for defining a structure. Key elements include:

- Buttons for 'Número de apoyos de la estructura' (set to 3) and 'Número de amortiguadores que hay en la estructura' (set to 1).
- A table for 'Desplazamiento en Y' and 'Desplazamiento en Z' with values 0, 0, 1, 0 for Y and 1, 1, 1, 1 for Z.
- A table for 'Constante de amortiguamiento del amortiguador' with values 1000, 0, 0 for bars 1, 2, and 3 respectively.
- A table for 'Defina las características de las barras' (highlighted with a red box) showing properties for three bars:

Barra	Modulo elástico (E)	Área	Momento de Inercia (I)	Densidad
1	7.0000e+10	7.8500e-05	9.8100e-10	2700
2	7.0000e+10	7.8500e-06	9.8100e-10	2700
3	7.0000e+10	7.8500e-06	9.8100e-10	2700

Figura 3.26: Comprobación de características de cada barra.

Como se puede comprobar todas las barras tienen idénticas características, exceptuando la barra uno, que tiene distinta área que las otras dos barras.

### 3.4.3. Definición del tipo de fuerza aplicada

Este sería el último paso de introducción de datos. Aquí el usuario deberá elegir si somete a la estructura a cargas estáticas (que no depende del tiempo) o a cargas dinámicas (que cambian con el tiempo)

Desde la ventana de Menú Principal, el usuario deberá de clicar sobre el botón Fuerzas Dinámicas E.R. (o E.A. si está analizando una estructura articulada) y se abrirá la siguiente ventana:

The screenshot shows a window titled "Fuerza\_Dinamica\_Reticulada" with a blue background. It contains several input fields and buttons:

- A button labeled "Número de nodos con una fuerza aplicada" at the top.
- A text input field below it.
- A button labeled "Indique el sentido de la fuerza" below the text input.
- A table with 6 columns: "Nodo", "Amplitud en eje X", "Amplitud en eje Y", "Momento", "Omega", and "Desfase". The table is currently empty.
- A button labeled "Intervalo de Tiempo" below the table.
- Two text input fields labeled "Tiempo inicial" and "Tiempo final" below the interval button.
- A button labeled "Aceptar" at the bottom.

Figura 3.27: Ventana donde se definirán los parámetros de la fuerza aplicada.

#### 3.4.3.1. Botón “Número de nodos con una fuerza aplicada”

Al pulsar este botón aparecerá una ventana como la siguiente:

The screenshot shows a window titled "Numero\_de\_Fuerzas\_ER" with a blue background. It contains the following elements:

- Text: "ESCRIBA EL NÚMERO DE NODOS EN EL QUE SE VA A APLICAR UNA FUERZA".
- A text input field.
- A button labeled "Aceptar" to the right of the input field.
- Text: "El número de nodos donde se va a aplicar una fuerza es el siguiente:".
- A text input field.
- Text: "Si es correcto pulse 'Cerrar'".
- A button labeled "Cerrar" at the bottom.

Figura 3.28: Ventana para indicar en cuantos nodos se va a aplicar una fuerza.

En la que el usuario deberá especificar en cuantos nodos de la estructura se va a aplicar la fuerza.

### 3.4.3.2. Botón “Indique el sentido de la fuerza”

Al pinchar sobre este botón aparecerá una ventana como la que se puede observar en la figura 3.29:

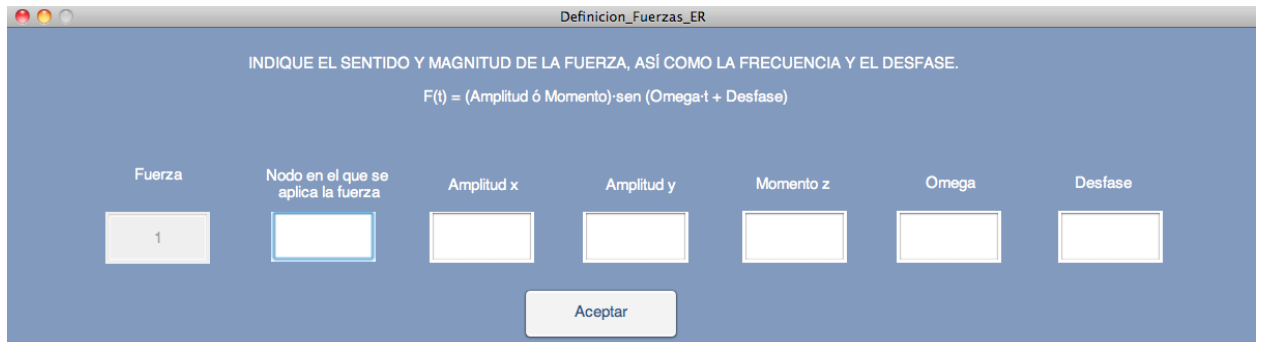


Figura 3.29: Ventana de definición de los parámetros de la fuerza aplicada en una estructura reticulada.

La caja de edición de texto no editable de la izquierda, sirve de recordatorio para el usuario, para saber que fuerza esta definiendo.

En el resto de cajas de edición de texto el usuario definirá en primer lugar el nodo en el que se aplica la fuerza.

La magnitud de la fuerza o el momento aplicado con signo negativo en caso de que el sentido de la fuerza o momento vayan en contra de los ejes de coordenadas predefinidos por el usuario.

Además deberá de definir los valores de la frecuencia y el desfase.

La ventana se abrirá y cerrará tantas veces como número de fuerzas halla indicado previamente el usuario.

### 3.4.3.3. Botón “Intervalo de Tiempo”

Se abrirá una ventana en la que el usuario definirá el intervalo de tiempo, en el que quiere analizar los desplazamientos y fuerzas de la estructura.

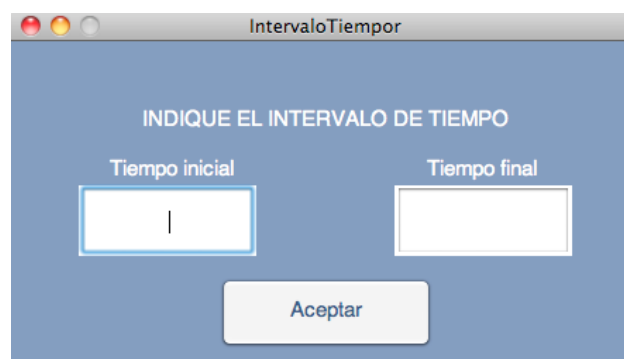


Figura 3.30: Ventana para introducir el intervalo de tiempo.

Como último paso se pulsará el botón Aceptar de la ventana Fuerza Dinámica Reticulada para que se guarden.

La ventana tendrá el siguiente aspecto, tras haberse introducido todos los datos:

The screenshot shows a window titled 'Fuerza\_Dinamica\_Reticulada' with a blue background. It contains several input fields and a table. At the top, there is a label 'Número de nodos con una fuerza aplicada' above a text box containing the number '1'. Below this is a label 'Indique el sentido de la fuerza' above a text box. In the center is a table with 7 columns: 'Nodo', 'Amplitud en eje X', 'Amplitud en eje Y', 'Momento', 'Omega', and 'Desfase'. The table has 4 rows of data. Below the table is a label 'Intervalo de Tiempo' above two text boxes: 'Tiempo inicial' containing '0' and 'Tiempo final' containing '0.06'. At the bottom is a large 'Aceptar' button.

Nodo	Amplitud en eje X	Amplitud en eje Y	Momento	Omega	Desfase
1	0	0	0	0	0
2	0	0	0	0	0
3	1000	0	0	1000	0
4	0	0	0	0	0

Figura 3.31: Ventana para introducir el intervalo de tiempo.

#### 3.4.4. Obtención de desplazamientos y fuerzas

Una vez introducidos todos los datos, se podrá clicar sobre el botón 1. *Cálculo de desplazamientos en dinámico E.R.* y saltará una ventana como la siguiente:

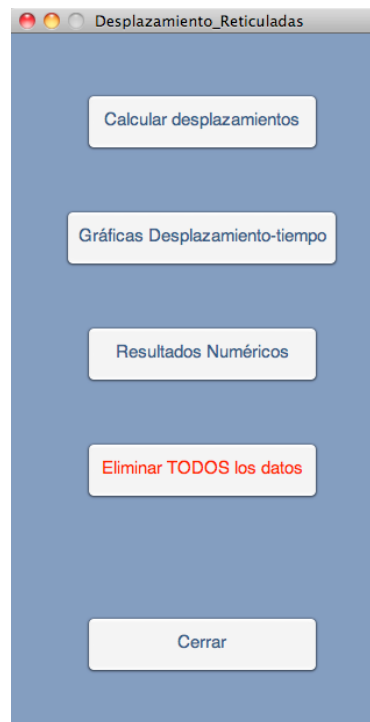


Figura 3.32: Ventana para calcular y comprobar resultados

El botón con letras rojas, sirve para eliminar los datos actuales del programa, en caso de que se quiera introducir unos nuevos.

En el momento en que se pulse el botón *Calcular desplazamientos*, Matlab procederá al cálculo de desplazamientos y velocidades de los nodos. Se podrán observar los resultados numéricos en una tabla al clicar *Resultados Numéricos*.

Así como gráficas en las que se podrá observar el desplazamiento del nodo en función del tiempo, al clicar el botón *Gráficas Desplazamiento-tiempo*.

Además se podrá comparar desplazamientos de distintos nodos de la misma estructura, e incluso, comparar desplazamientos de nodos de distintas estructuras.

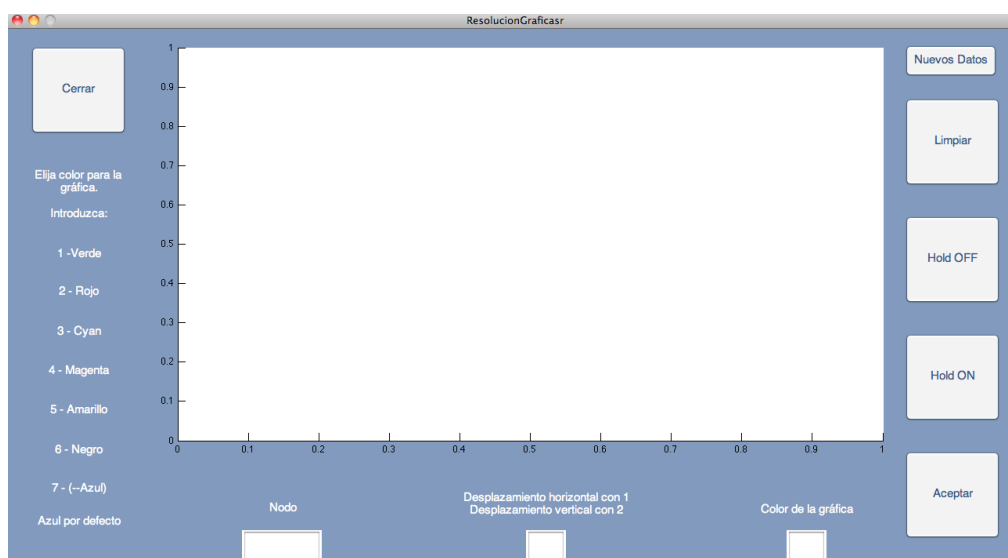


Figura 3.33: Ventana donde se grafican los desplazamientos frente al tiempo

Como se puede observar en esta ventana se dispone de varios botones y cada uno con una función determinada:

El usuario indicará el número del nodo que quiera visualizar en la gráfica en la caja de abajo a la izquierda, se indicará si se quiere comprobar el desplazamiento horizontal o vertical en la caja de en medio y opcionalmente, se podrá elegir el color que quiere que tenga la gráfica. Introducidos estos datos, se pulsará Aceptar e instantáneamente aparecerá la gráfica solicitada.

En caso de querer comparar la gráfica con alguna otra, se pulsara el botón Hold On, se introducirán los datos y se volverá a pulsar Aceptar, pudiéndose comparar todas las gráficas que se deseen.

Para que estas gráficas se dejen de superponer se utilizará el botón Hold Off. Y si se quiere eliminar cualquier gráfica para dejar la pantalla en blanco se empleará el botón Limpiar.

En caso de quererse comparar la gráfica, con cualquier otra gráfica de otra estructura, se pulsara Hold On y posteriormente el botón Nuevos Datos.

En ese momento se abandonara esa ventana, pero sin cerrarse, y se procederá a la introducción de los datos de la nueva estructura, seguido del posterior cálculo de desplazamientos y por último, pudiendo comparar ambas estructuras en la ventana de gráficas de desplazamientos frente al tiempo.

Si no se desea realizar ninguna otra operación, se pulsara Cerrar y se procederá al cálculo de las reacciones de la estructura.

Para las reacciones de la estructura habrá que pulsar el botón 2. *Cálculo de fuerzas en dinámico* y las ventanas y procedimiento de comprobación de resultados será el mismo que para los desplazamientos, solo que en este caso los resultados serán las fuerzas resultantes de la estructura.

En el caso de desplazamientos y fuerzas en estático, la ventana que saltara será la siguiente:

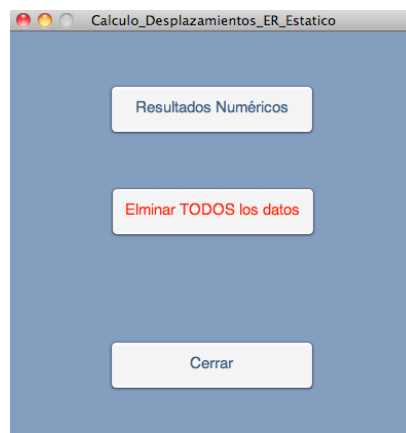


Figura 3.34: Ventana para obtención de resultados de desplazamientos en estructuras reticuladas.

En este caso, se observarán los resultados numéricos de los desplazamientos y reacciones de la estructura en tablas.

### **3.4.5. Obtención de valores y modos propios de la estructura.**

Introducidos los datos que definen la estructura, solo habrá que clicar el botón de *Frecuencias y modos propios* del Menú Principal, y automáticamente, saltará una ventana con un tabla, en la que, en la primera columna se mostrará el valor de cada uno de los valores propios y a su derecha, los modos propios asociados a cada valor.

Cada uno de los valores del vector de modos propios, estará asociado a un grado de libertad de la estructura (por ejemplo, el desplazamiento horizontal del nodo uno), y cuanto más cercano sea este valor a uno, indicará que el desplazamiento será mayor en ese sentido del nodo, si se aplica esa frecuencia angular.

## **3.5. Estructura del programa**

Para la realización del programa UCMM ha sido necesaria la creación de distintas funciones (ficheros \*.m) para cada uno de los cálculos descritos en el apartado 3.3.

Con ello se consigue:

- Simplificar las líneas de programación.
- Reducir el tiempo de cálculo computacional.
- Facilitar la localización de errores y modificación de los ficheros.

Los ficheros que permiten el cálculo de estructuras son 18, y se encuentran en los anexos del 1 al 5. Son los siguientes:

### **Cálculo de matrices de rigidez y amortiguamiento**

- 1- Matriz de rigidez barras en ejes globales EA.
- 2- Matriz de rigidez barras en ejes globales ER.
- 3- Matriz de rigidez muelle en ejes globales EA.
- 4- Matriz de rigidez muelle en ejes globales ER.
- 5- Matriz de amortiguamientos en ejes globales EA.
- 6- Matriz de amortiguamientos en ejes globales ER.

### **Ficheros para la obtención de desplazamientos y fuerzas**

- 7- Cálculos previos desplazamientos ER.
- 8- Obtención de desplazamientos para estructuras reticuladas en dinámico (ode45).
- 9- Cálculos previos desplazamientos EA.
- 10- Obtención de desplazamientos para estructuras articuladas en dinámico (ode45).
- 11- Obtención de fuerzas y reacciones para estructuras reticuladas en dinámico.
- 12- Obtención de fuerzas y reacciones para estructuras articuladas en dinámico.
- 13- Obtención de desplazamientos para estructuras reticuladas en estático.
- 14- Obtención de desplazamientos para estructuras articuladas en estático.
- 15- Obtención de fuerzas y reacciones para estructuras reticuladas en estático.
- 16- Obtención de fuerzas y reacciones para estructuras articuladas en estático.
- 17- Valores y modos propios estructura articulada
- 18- Valores y modos propios estructura reticulada

Los ficheros de la interfaz gráfica son 62, y se encuentran en el anexo 6. Son los siguientes:



- 1- Menú Principal.
- 2- Definición estructura articulada.
- 3- Definición estructura reticulada.
- 4- Definición de los nodos de cada barra ER.
- 5- Definición de los nodos de cada barra EA.
- 6- Introducir Número de Nodos EA.
- 7- Introducir Número de Nodos ER.
- 8- Valor de la constante de rigidez ER.
- 9- Valor de la constante de rigidez EA.
- 10-Coordenadas de cada de nodo ER.
- 11-Coordenadas de cada de nodo EA.
- 12- Tipo de apoyo ER.
- 13-Tipo de apoyo EA.
- 14- Valor de la constante de amortiguamiento ER.
- 15-Valor de la constante de amortiguamiento EA.
- 16- Especificación de las características de las secciones ER.
- 17-Especificación de las características de las secciones EA.
- 18-Número de barras de la EA.
- 19-Número de barras de la ER.
- 20-Número de barras de la EA.
- 21-Número de apoyos de la ER.
- 22-Número de apoyos de la EA.
- 23- Número de muelles que tiene la EA.
- 24-Número de muelles que tiene la ER.
- 25- Número de amortiguadores que tiene la EA.
- 26-Número de amortiguadores que tiene la ER.
- 27-Definición de fuerzas estáticas para EA.
- 28-Definición de fuerzas estáticas para ER.
- 29-Número de fuerzas estáticas aplicadas en la ER.
- 30-Número de fuerzas estáticas aplicadas en la EA.
- 31-Características fuerzas estáticas ER.
- 32-Características fuerzas estáticas EA.
- 33-Ventana de fuerzas dinámicas aplicadas a ER.
- 34-Ventana de fuerzas dinámicas aplicadas a EA.
- 35-Número de fuerzas dinámicas aplicadas a la ER.
- 36-Número de fuerzas dinámicas aplicadas a la EA.
- 37-Definición fuerzas dinámicas aplicadas a la ER.
- 38-Definición fuerzas dinámicas aplicadas a la EA.
- 39-Intervalo de tiempo en EA.
- 40-Intervalo de tiempo en ER.
- 41-Ventana de resultados de valores y modos propios EA
- 42-Ventana de resultados de valores y modos propios ER
- 43-Resultados desplazamientos en estático ER.
- 44-Resultados desplazamientos en estático EA.
- 45-Resultados numéricos desplazamientos en estático ER.
- 46-Resultados numéricos desplazamientos en estático EA.
- 47-Ventana resultados fuerzas estático ER.
- 48-Ventana resultados fuerzas estático EA.
- 49-Resultados numéricos fuerzas estáticas ER.
- 50-Resultados numéricos fuerzas estáticas EA.

- 51-Ventana resultados desplazamientos dinámicos ER.
- 52-Ventana resultados desplazamientos dinámicos EA.
- 53-Resultados numéricos desplazamientos dinámicos ER.
- 54-Resultados numéricos desplazamientos dinámicos EA.
- 55-Resultados gráficas desplazamientos dinámicos ER.
- 56-Resultados gráficas desplazamientos dinámicos EA.
- 57-Ventana resultados fuerzas dinámicas ER.
- 58-Ventana resultados fuerzas dinámicas EA.
- 59-Resultados gráficas fuerzas dinámico ER.
- 60-Resultados gráficas fuerzas dinámico EA.
- 61-Resultados numéricos fuerzas dinámicas ER.
- 62-Resultados numéricos fuerzas dinámicas EA.

## 4. Validación del programa

Para validar el programa se analizarán tres estructuras distintas, dos articuladas y una reticulada. Se calcularán los desplazamientos y reacciones en condiciones estáticas mediante tres métodos:

- Cálculo analítico.
- Cálculo mediante el programa Ed-Tridim.
- Cálculo mediante el programa UCMM.

Obtenidos los resultados y realizando una comparación y verificación de que los valores obtenidos mediante los distintos procedimientos son similares, quedará validado el buen funcionamiento del programa en condiciones estáticas.

Posteriormente, se compararán los valores de desplazamientos y reacciones obtenidos en el caso estático, con el caso dinámico mediante gráficas. Comprobándose la coherencia de los resultados obtenidos y no por ello validando el programa en condiciones dinámicas (aunque los resultados obtenidos en condiciones dinámicas figuren en este capítulo del documento).

Se sustituirán barras de la estructura por muelles y amortiguadores, y se observará la respuesta de la estructura en gráficas, donde se observará como se reducen o aumentan los desplazamientos y fuerzas en los nodos.

## 4.1. Estructura 1

### 4.1.1. Obtención de desplazamientos y reacciones en estático

Se obtendrán los desplazamientos, reacciones, valores y modos propios de una estructura articulada mediante distintos procedimientos, para comparar los resultados obtenidos y verificar que el programa da resultados coherentes.

#### 4.1.1.1. Resolución analítica

Para la estructura articulada que se muestra en la figura 4.1, constituida por barras de aluminio con la misma sección, se desea obtener:

- Desplazamiento de los nodos.
- Reacciones.

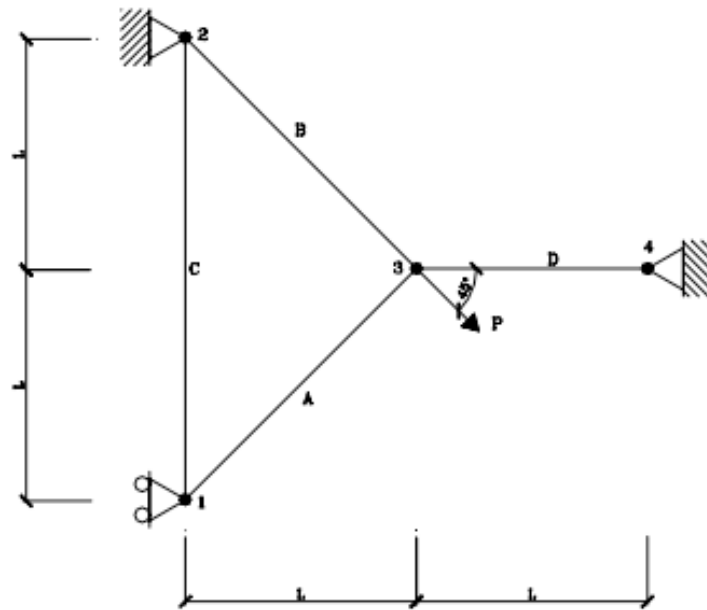


Figura 4.1: Estructura articulada 1

Los datos para la resolución del problema se encuentran en la siguiente tabla:

Longitud (m)	Densidad (Kg/m <sup>3</sup> )	Modulo de elasticidad (N/m <sup>2</sup> )	Fuerza (N)	Área (m <sup>2</sup> )
0,5	2700	$70 \cdot 10^9$	1000	$7.854 \cdot 10^{-5}$

Tabla 4.1: Datos para la resolución del estructura uno.

### RESOLUCIÓN

Se tiene una estructura configurada por distintas barras y orientaciones. En la siguiente tabla se muestra la conectividad de los nodos de cada barra, y el ángulo que forman con los ejes globales:

Barras	Nodo de menor numeración	Nodo de mayor numeración	Ángulo entre ejes locales y globales (°)
A	1	3	45
B	2	3	315
C	1	2	90
D	3	4	0

Tabla 4.2: Definición de cada barra y ángulo que forman con ejes globales.

La matriz de rigidez de la estructura completa ( $[K]$ ), cuyo procedimiento de cálculo se explica en el apartado 2.1.2.2.5. de antecedentes, será de la siguiente forma:

$$\begin{bmatrix} (K_{11}^A)_{11} + (K_{11}^C)_{11} & (K_{12}^C)_{12} & (K_{12}^A)_{13} & 0 \\ (K_{21}^C)_{21} & (K_{11}^B)_{22} + (K_{22}^C)_{22} & (K_{12}^B)_{23} & 0 \\ (K_{21}^A)_{31} & (K_{21}^B)_{32} & (K_{22}^A)_{33} + (K_{22}^B)_{33} + (K_{11}^D)_{33} & (K_{12}^D)_{34} \\ 0 & 0 & (K_{21}^D)_{43} & (K_{22}^D)_{44} \end{bmatrix} \quad (4.1)$$

Se tendrá que calcular la matriz de rigidez de los distintos elementos en coordenadas globales a partir de su expresión en locales y de las matrices de transformación de coordenadas. El sentido de los ejes en locales es del nudo inicial al final de cada elemento, según la tabla anterior.

Al tratarse de una estructura con nudos articulados, los movimientos permitidos en cada uno de ellos son únicamente los acortamientos o alargamientos en la dirección de los esfuerzos axiales (la propia dirección de la barra), por tanto la matriz de rigidez elemental de cada barra será de la forma:

$$\begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \quad (4.2)$$

**- BARRA A**

Matriz de la barra =

$$\begin{bmatrix} \frac{EA}{L\sqrt{2}} & -\frac{EA}{L\sqrt{2}} \\ -\frac{EA}{L\sqrt{2}} & \frac{EA}{L\sqrt{2}} \end{bmatrix} \quad (4.3)$$

Longitud =  $L\sqrt{2}$ ,  $\alpha = 45^\circ$ , con lo que la matriz en coordenadas globales es:

$$K^A = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \cdot \frac{EA}{L\sqrt{2}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \frac{EA\sqrt{2}}{4L} \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad (4.4)$$

**- BARRA B**

Longitud =  $L\sqrt{2}$ ,  $\alpha = 315^\circ$ , matriz de la barra igual a la anterior (4.3), con lo que la matriz de la barra B en globales es:

$$K^B = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot \frac{EA}{L\sqrt{2}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \frac{EA\sqrt{2}}{4L} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (4.5)$$

**- BARRA C**

Matriz de la barra será la (4.2), longitud =  $2L$ ,  $\alpha = 90^\circ$ , con lo que la matriz en coordenadas globales es:

$$K^C = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \frac{EA}{2L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{EA}{2L} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (4.6)$$

**- BARRA D**

Longitud =  $L$ ,  $\alpha = 0^\circ$ , matriz de la barra igual que la anterior, con lo que la matriz de la barra D en globales es:

$$K^D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.7)$$

La matriz de rigidez de la estructura completa se obtiene ensamblando las de cada barra:

$$K = \frac{EA}{L} \begin{bmatrix} \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} + \frac{1}{2} & 0 & -\frac{1}{2} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} + \frac{1}{2} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 1 + \frac{\sqrt{2}}{2} & 0 & -1 & 0 \\ -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.8)$$

Como se explico en el apartado 2.1.2.2.7. de antecedentes, el siguiente paso será la obtención de la matriz de rigidez reducida ( $[K^*]$ ). Teniendo en cuenta los grados de libertad restringidos en los diferentes nudos, se pueden suprimir las filas y columnas correspondientes, que son las 1, 3, 4, 7 y 8 (movimiento x del nudo 1, y los dos movimientos x e y de los nudos 2 y 4). Obteniéndose la siguiente matriz de rigidez reducida ( $K^*$ ):

$$K^* = \begin{bmatrix} \frac{\sqrt{2}}{4} + \frac{1}{2} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} + 1 & 0 \\ -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} \end{bmatrix} \quad (4.9)$$

Y queda el siguiente sistema de ecuaciones:

$$\frac{P\sqrt{2}}{2} \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} \frac{\sqrt{2}}{4} + \frac{1}{2} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} + 1 & 0 \\ -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} \end{bmatrix} \cdot \begin{bmatrix} \overline{u}_1 \\ \overline{u}_3 \\ \overline{v}_3 \end{bmatrix} \quad (4.10)$$



Resolviendo el sistema se obtiene que los desplazamientos son:

$$\begin{aligned}\bar{v}_1 &= -0,343 \frac{PL}{EA} = -3,12 \cdot 10^{-5} m \\ \bar{u}_3 &= 0,343 \frac{PL}{EA} = 3,12 \cdot 10^{-5} m \\ \bar{v}_3 &= -1,171 \frac{PL}{EA} = -1,065 \cdot 10^{-4} m\end{aligned}\quad (4.11)$$

El cálculo de las reacciones es inmediato sustituyendo el valor obtenido de los desplazamientos en el sistema de ecuaciones original. Al tener solamente reacciones en los nudos 1 (en dirección x únicamente), 2 y 4, sólo son necesarias las ecuaciones correspondientes a esos nudos y direcciones. De la matriz de rigidez de la estructura se tiene:

$$\begin{bmatrix} H_1 \\ H_2 \\ V_2 \\ H_4 \\ V_4 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} + \frac{1}{2} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{PL}{EA} = \begin{bmatrix} 0 \\ -0,343 \\ 0 \\ 0,343 \\ -1,171 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,171P \\ -0,53P \\ 0,705P \\ -0,343P \\ 0 \end{bmatrix}\quad (4.12)$$

$$\begin{aligned}V_1 &= 171N \\ H_2 &= -530N \\ V_2 &= 705N \\ H_4 &= -343N \\ V_4 &= 0N\end{aligned}\quad (4.13)$$

Las soluciones se pueden observar en la siguiente tabla:

Desplazamientos (m)	Reacciones y Fuerzas (N)
$u_1 = -3,120 \cdot 10^{-5}$	$H_1 = 0$
$v_1 = 0$	$V_1 = 171$
$u_2 = 0$	$H_2 = -530$
$v_2 = 0$	$V_2 = 705$
$u_3 = 3,120 \cdot 10^{-5}$	$FH_3 = 707,1$
$v_3 = -1,065 \cdot 10^{-4}$	$FV_3 = -707,1$
$u_4 = 0$	$H_4 = -343$
$v_4 = 0$	$V_4 = 0$

Tabla 4.3: Resultados de desplazamientos y reacciones de la estructura 1, obtenidos mediante cálculo analítico.

#### 4.1.1.2. Resolución con Ed-Tridim

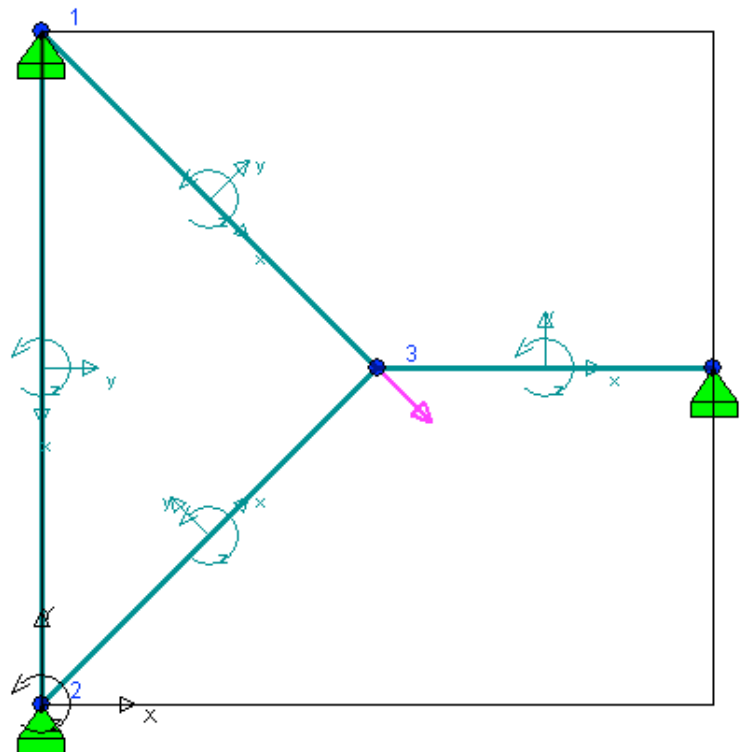


Figura 4.2: Estructura para resolución mediante Ed-Tridim

Resultados obtenidos con Ed-Tridim:

Desplazamientos (m)	Reacciones y Fuerzas (N)
$u_1 = -3,120 \cdot 10^{-5}$	$H_1 = 5,680 \cdot 10^{-14}$
$v_1 = 0$	$V_1 = 172$
$u_2 = 0$	$H_2 = -536$
$v_2 = 0$	$V_2 = 707$
$u_3 = 3,120 \cdot 10^{-5}$	$FH_3 = 707,1$
$v_3 = -1,070 \cdot 10^{-4}$	$FV_3 = -707,1$
$u_4 = 0$	$H_4 = -343$
$v_4 = 0$ m	$V_4 = 0$ N

Tabla 4.4: Resultados de desplazamientos y reacciones de la estructura 1, obtenidos mediante Ed-Tridim.

#### 4.1.1.3. Resolución con UCMM

Desplazamientos (m)	Reacciones y Fuerzas (N)
$u_1 = -3,120 \cdot 10^{-5}$	$H_1 = 5,684 \cdot 10^{-14}$
$v_1 = 0$	$V_1 = 171.547$
$u_2 = 0$	$H_2 = -535.453$
$v_2 = 0$	$V_2 = 707$
$u_3 = 3,120 \cdot 10^{-5}$	$FH_3 = 707,100$
$v_3 = -1,065 \cdot 10^{-4}$	$FV_3 = -707,100$
$u_4 = 0$	$H_4 = -343,094$
$v_4 = 0$	$V_4 = 0$

Tabla 4.5: Resultados de desplazamientos y reacciones de la estructura 1, obtenidos mediante UCMM.

En las siguientes dos tablas se pueden comparar los resultados obtenidos mediante los tres procedimientos. En el caso de los desplazamientos:

Cálculo analítico (m)	Ed-Tridim (m)	UCMM (m)
$u_1 = -3,120 \cdot 10^{-5}$	$u_1 = -3,120 \cdot 10^{-5}$	$u_1 = -3,120 \cdot 10^{-5}$
$v_1 = 0$	$v_1 = 0$	$v_1 = 0$
$u_2 = 0$	$u_2 = 0$	$u_2 = 0$
$v_2 = 0$	$v_2 = 0$	$v_2 = 0$
$u_3 = 3,120 \cdot 10^{-5}$	$u_3 = 3,120 \cdot 10^{-5}$	$u_3 = 3,120 \cdot 10^{-5}$
$v_3 = -1,065 \cdot 10^{-4}$	$v_3 = -1,070 \cdot 10^{-4}$	$v_3 = -1,065 \cdot 10^{-4}$
$u_4 = 0$	$u_4 = 0$	$u_4 = 0$
$v_4 = 0$	$v_4 = 0$	$v_4 = 0$

Tabla 4.6: Tabla comparativa de resultados de desplazamientos en la estructura 1, obtenidos mediante cálculo analítico, Ed-Tridim y UCMM.

Para las reacciones y fuerzas:

Cálculo analítico (N)	Ed-Tridim (N)	UCMM(N)
$H_1 = 0$	$H_1 = 5,680 \cdot 10^{-14}$	$H_1 = 5,684 \cdot 10^{-14}$
$V_1 = 171$	$V_1 = 172$	$V_1 = 171.547$
$H_2 = -530$	$H_2 = -536$	$H_2 = -535.453$
$V_2 = 705$	$V_2 = 707$	$V_2 = 707$
$FH_3 = 707,100$	$FH_3 = 707,100$	$FH_3 = 707,100$
$FV_3 = -707,100$	$FV_3 = -707,100$	$FV_3 = -707,100$
$H_4 = -343$	$H_4 = -343$	$H_4 = -343,094$
$V_4 = 0$	$V_4 = 0$	$V_4 = 0 \text{ N}$

Tabla 4.7: Tabla comparativa de resultados de reacciones en la estructura 1, obtenidos mediante cálculo analítico, Ed-Tridim y UCMM.

### 4.1.2. Valores y modos propios de la estructura

A continuación se realizará un cálculo de los valores y modos propios de la estructura 1 mediante dos procedimientos, el primero será mediante cálculo analítico, y el segundo mediante UCMM. Con ello quedará demostrado que el UCMM realiza los cálculos correctamente.

#### 4.1.2.1. Cálculo analítico

En primer lugar se realizará el cálculo de los valores propios de la estructura aplicando la siguiente fórmula:

$$|[K^*] \cdot [M^*]^{-1} - \omega \cdot I| = 0 \quad (4.14)$$

Donde  $[K^*]$  y  $[M^*]$  son las matrices de rigidez y masas reducidas de la estructura.

Siendo la matriz de rigidez reducida la que se muestra en (4.9) y la de masas reducida:

$$M = m \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} \Rightarrow M^* = m \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{3}{2} \end{bmatrix} \quad (4.15)$$

Introduciendo el valor de las matrices (4.9) y (4.15) en la ecuación (4.14), y realizando los cálculos, se obtienen tres valores de frecuencia propia de la estructura 1:

$$\begin{aligned} \omega_1 &= 4640 \text{ Hz} \\ \omega_2 &= 7700 \text{ Hz} \\ \omega_3 &= 10000 \text{ Hz} \end{aligned} \quad (4.16)$$

Con los valores de las frecuencias obtenidas anteriormente en (4.16), se podrán obtener los modos propios asociados a cada valor de frecuencia, a partir de la fórmula (4.17):

$$([K^*] - \omega^2 \cdot [M^*]) \cdot \{x\} = 0 \quad (4.17)$$

Siendo el primer valor de cada vector, el asociado al desplazamiento vertical del nodo uno ( $v_1$ ), el segundo valor, el asociado al desplazamiento horizontal del nodo tres ( $u_3$ ), y el tercer valor, el asociado al desplazamiento vertical del nodo tres ( $v_3$ ).

$$\begin{aligned}\{x_1\} &= (-0,648 \ -0,180 \ -0,772) \\ \{x_2\} &= (-0,631 \ -0,405 \ -0,651) \\ \{x_3\} &= (0,398 \ -0,909 \ -0,138)\end{aligned}\quad (4.18)$$

En la siguiente tabla se muestra una comparación de los valores (4.16) de las frecuencias propias obtenidas mediante cálculo analítico, con los obtenidos mediante UCMM:

	Frecuencias propias obtenidas mediante cálculo analítico (Hz)	Frecuencias propias obtenidas mediante UCMM(Hz)
Frecuencia propia 1 ( $\omega_1$ )	4640	4640
Frecuencia propia 2 ( $\omega_2$ )	7700	7693
Frecuencia propia 3 ( $\omega_3$ )	10000	10095

Tabla 4.8 : Tabla comparativa de resultados de frecuencias propias obtenidas mediante cálculo analítico y UCMM.

Exactamente igual que para las frecuencias propias, en la siguiente tabla se comparan los valores de los modos propios asociados a cada frecuencia propia, calculados analíticamente y mediante el programa.

	Frecuencias propias	Desplazamiento vertical del nodo 1 ( $v_1$ )	Desplazamiento horizontal del nodo 3 ( $u_3$ )	Desplazamiento vertical del nodo 3 ( $v_3$ )
Modos propios obtenidos mediante cálculo analítico	$\omega_1$	-0,648	-0,180	-0,772
	$\omega_2$	-0,631	-0,405	-0,651
	$\omega_3$	0,398	-0,909	-0,138
Modos propios obtenidos mediante UCMM	$\omega_1$	-0,605	-0,183	-0,775
	$\omega_2$	-0,636	-0,410	0,654
	$\omega_3$	0,399	-0,907	-0,135

Tabla 4.9 : Tabla comparativa de resultados de modos propios de cada frecuencia, obtenidos mediante cálculo analítico y UCMM.

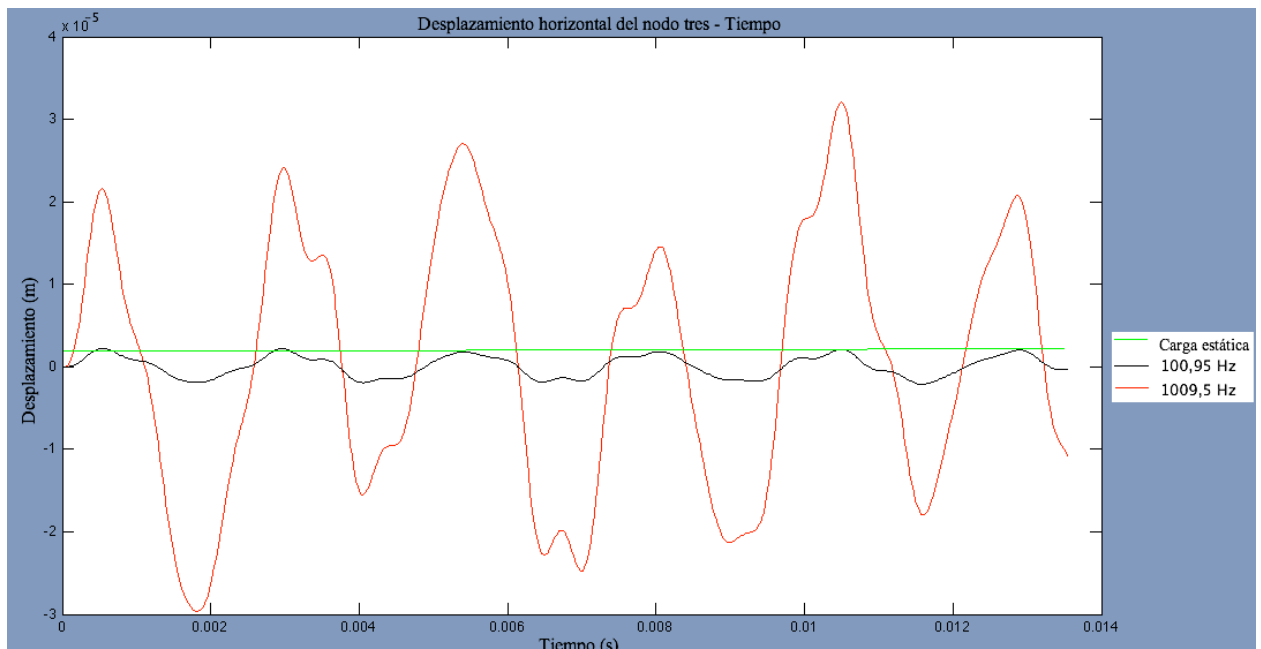
### 4.1.3. Obtención de desplazamientos y reacciones en dinámico

En este apartado se comparará el comportamiento del nodo tres de la estructura y reacción vertical del nodo dos, cuando se aplique una carga dinámica con una fuerza de excitación de 1000 N, misma dirección y sentido que la aplicada en el caso estático (figura 4.1), y un valor de frecuencia angular a la que se dará tres valores distintos. Conocido el valor de la frecuencia propia que mas afecta a la estructura ( $\omega_3$ ) de la tabla 4.7, se estudiarán tres casos en los que se tendrán los siguientes valores de frecuencias angulares:

- a) Frecuencia 1  $\rightarrow \omega = \omega_3/100 = 100,95$  Hz
- b) Frecuencia 2  $\rightarrow \omega = \omega_3/10 = 1009,5$  Hz
- c) Frecuencia 3  $\rightarrow \omega = \omega_3 = 10095$  Hz

#### 4.1.3.1. Desplazamientos en el nodo tres

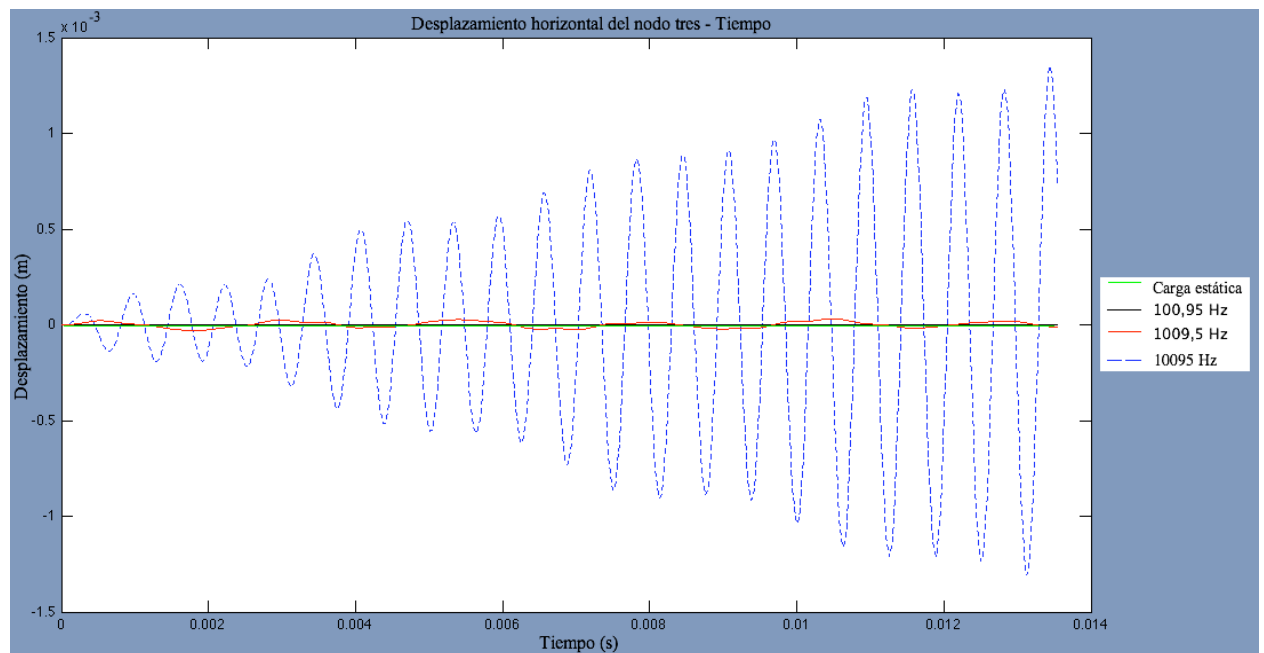
En la siguiente gráfica se muestra una comparación del desplazamiento del nodo tres en sentido horizontal, cuando se aplica una carga estática, y carga dinámica. Las cargas dinámicas aplicadas estarán excitadas con dos frecuencias distintas, ambas de valores alejados al valor de la frecuencia propia ( $\omega_3$ ), para evitar obtener grandes desplazamientos. Un valor de frecuencia angular será de 100,95 HZ y otro será de 1009,5 Hz.



Gráfica 4.1 : Desplazamiento horizontal del nodo tres frente al tiempo, aplicándose carga estática, y carga dinámica para  $\omega=100,95$  Hz y  $\omega=1009,5$  Hz.

Como se ha podido observar en la gráfica 4.1, ha medida que el valor de la frecuencia angular de la fuerza dinámica aplicada se aproxima a valores de frecuencias propias de la estructura, el valor del desplazamiento obtenido aumenta, alejándose del valor obtenido cuando se aplica una carga estática.

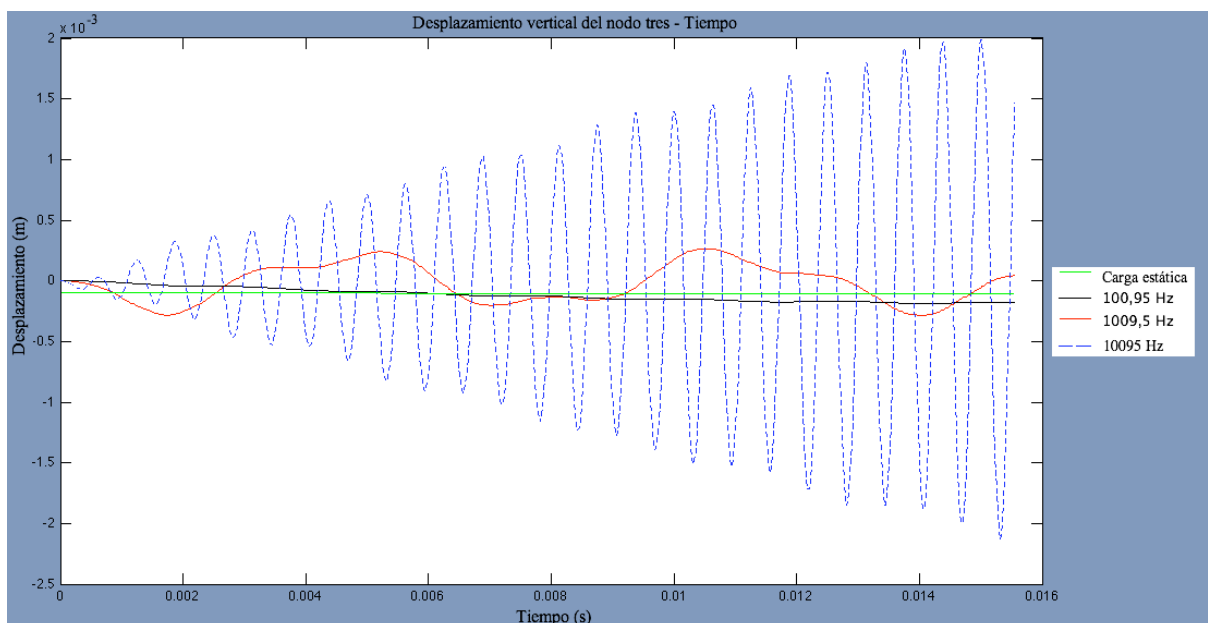
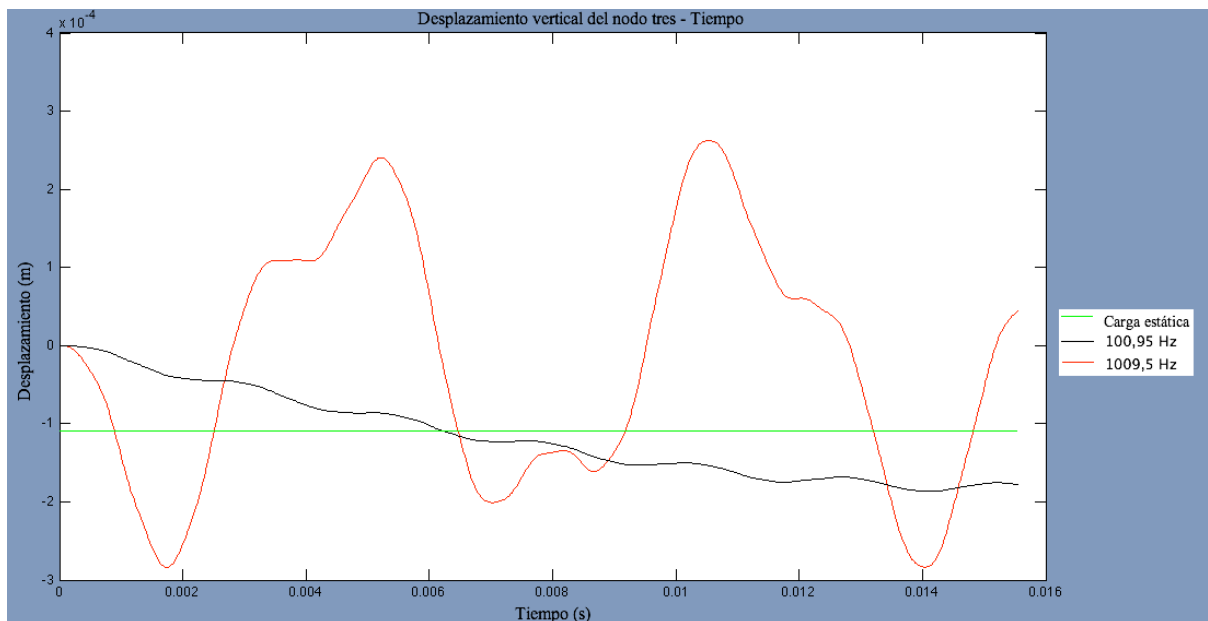
En la siguiente gráfica se mostraran los mismos valores de los desplazamientos representados en la grafica 4.1 y los obtenidos cuando se excita la carga dinámica con la frecuencia propia de la estructura.



Grafica 4.2 : Desplazamiento horizontal del nodo tres frente al tiempo, aplicándose carga estática, y carga dinámica para  $\omega=100,95$  Hz,  $\omega=1009,5$  Hz y  $\omega=10095$  Hz.

Se puede apreciar que con la frecuencia propia, los valores de desplazamiento obtenidos son tan grandes, que al compararlos con las otras frecuencias sus desplazamientos son casi nulos.

A continuación se representarán los valores de desplazamientos verticales obtenidos del nodo tres, para las mismas condiciones que las graficas 4.1 y 4.2.

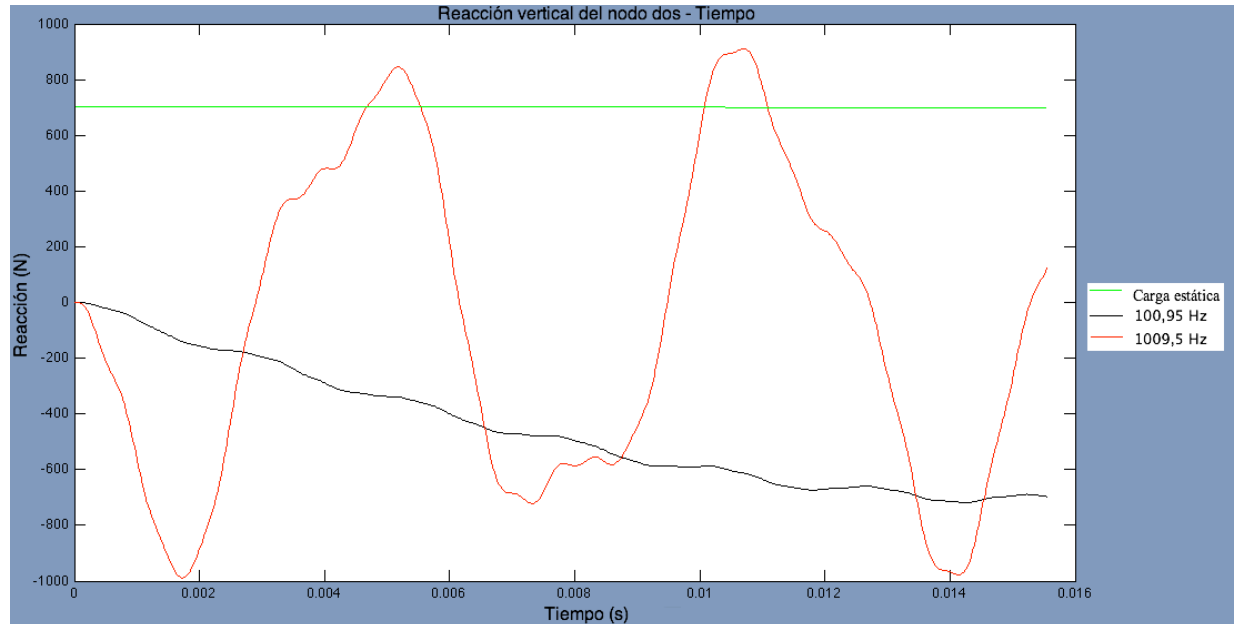


Para las gráficas 4.3 y 4.4, las conclusiones serían las mismas que para las gráficas 4.1 y 4.2 respectivamente. Destacando que se produce un mayor desplazamiento en sentido vertical.



#### 4.1.3.2. Reacción vertical del nodo dos

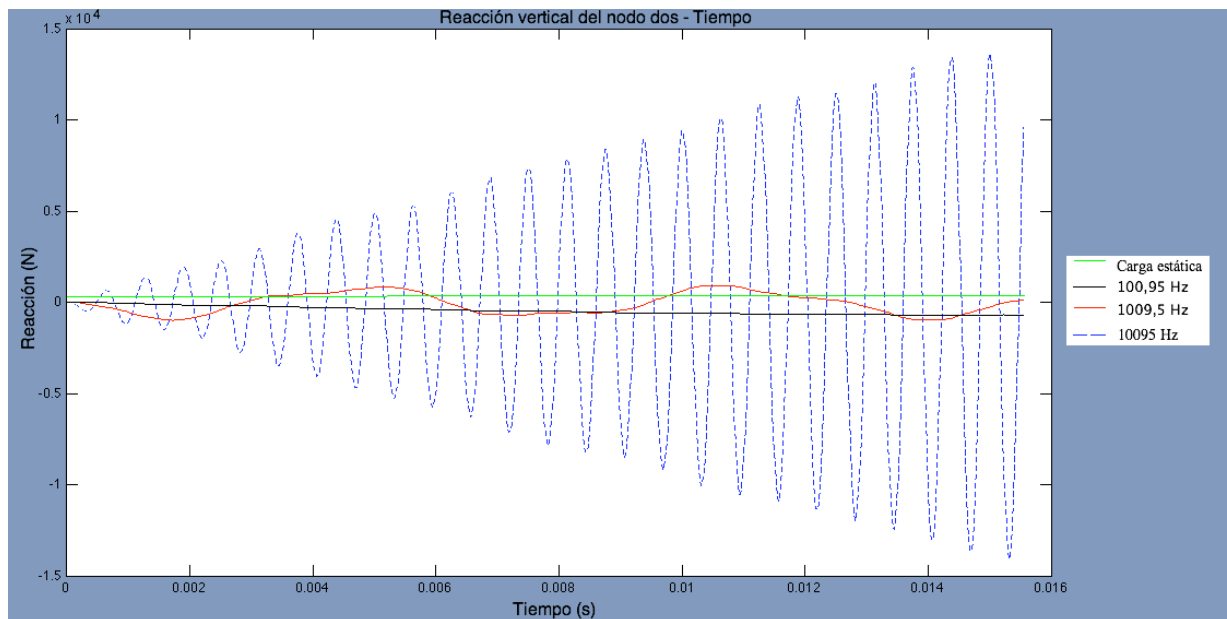
En este apartado se mostrarán los valores que toman la reacción vertical del nodo dos, cuando se somete a la estructura a carga estática, y carga dinámica con dos valores distintos de frecuencia angular.



Grafica 4.5 : Reacción vertical del nodo dos frente al tiempo, aplicándose carga estática, y carga dinámica para  $\omega=100,95$  Hz y  $\omega=1009,5$  Hz.

Se puede observar que los valores de las reacciones obtenidas cuando se aplica una frecuencia angular alejada de los valores de las frecuencias propias de la estructura, son parecidos al valor que se obtiene en el caso estático. Y que aumentan a medida que se aplica una carga dinámica con valores de frecuencia cercanos al de las frecuencias propias.

En la siguiente se comparan los resultados obtenidos en la gráfica 4.5, con los obtenidos para la frecuencia propia.



Grafica 4.6 : Reacción vertical del nodo dos frente al tiempo, aplicándose carga estática, y carga dinámica para  $\omega=100,95$  Hz,  $\omega=1009,5$  Hz y  $\omega=10095$  Hz.

Para una carga dinámica excitada con la frecuencia propia, los valores de la reacción aumentan con el tiempo.

### 4.1.4. Introducción de muelles y amortiguadores en la estructura

Se introducirán muelles y amortiguadores de distintos valores en la estructura, para observar como influyen éstos en los desplazamientos de los nodos.

#### 4.1.4.1. Introducción de un muelle en la estructura

En este apartado se sustituirá la barra D de la estructura, por muelles con distintas constantes de rigidez, y se someterá a la estructura a una carga dinámica, con una fuerza de excitación de 1000 N y una frecuencia angular  $\omega=100,95$  Hz.

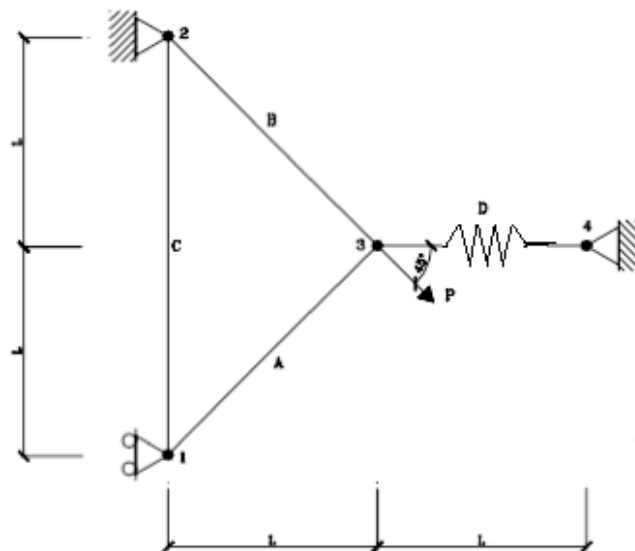


Figura 4.3: Estructura 1 con muelle.

Para tener como referencia un valor de constante de rigidez del muelle, se calculará la rigidez de las barras de la estructura, y a partir del valor obtenido, se irá variando su valor. El cálculo se puede observar a continuación:

$$F = K \cdot X \quad \rightarrow \quad N = K \cdot \Delta L \quad (4.18)$$

Para una barra se tendría:

$$\left. \begin{aligned} \sigma &= E \cdot \varepsilon \\ N &= E \cdot A \cdot \varepsilon \\ N &= E \cdot A \cdot \frac{\Delta L}{L} \end{aligned} \right\} \Rightarrow N = \frac{EA}{L} \Delta L \quad (4.19)$$

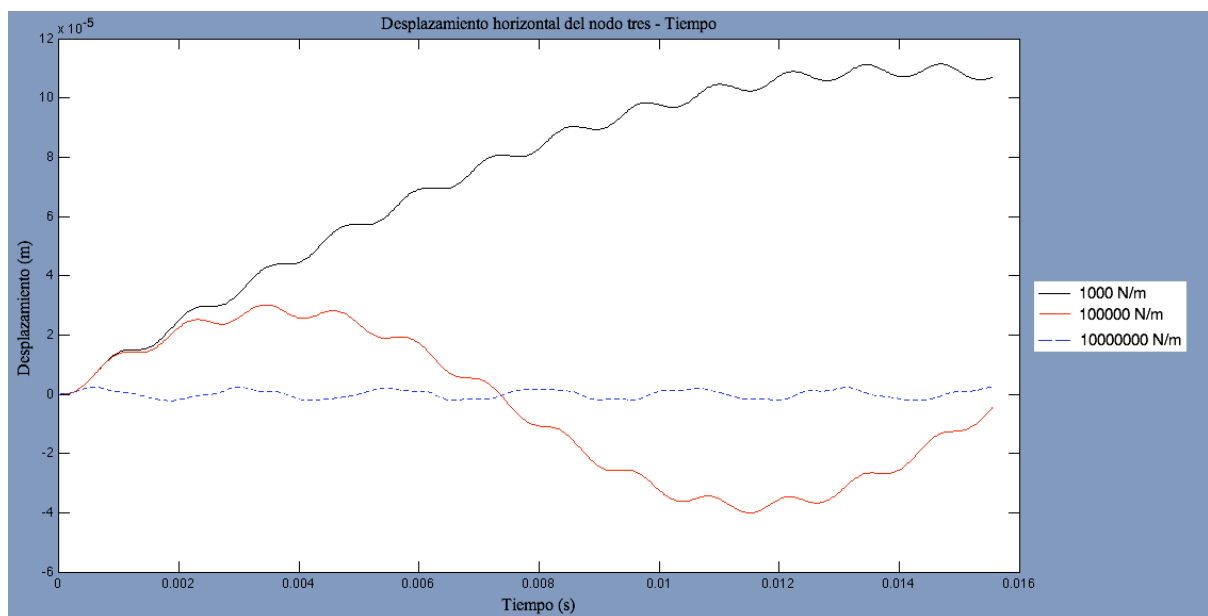
Por tanto:

$$K_{eq} = EA/L \rightarrow K_{eq} = 70 \cdot 10^9 \cdot 7,854 \cdot 10^{-5} / 0,5 = 10 \cdot 10^6 \text{ N/m} \quad (4.20)$$

El valor de constante de rigidez del muelle equivalente a las barras de la estructura sería el que se muestra en (4.20). Se tomará como referencia ese valor para seleccionar tres muelle:

- a) Muelle 1  $\rightarrow K_{\text{muelle1}} = K_{\text{eq}} = 10 \cdot 10^2 \text{ N/m}$
- b) Muelle 2  $\rightarrow K_{\text{muelle2}} = K_{\text{eq}}/100 = 10 \cdot 10^4 \text{ N/m}$
- c) Muelle 3  $\rightarrow K_{\text{muelle3}} = K_{\text{eq}}/10000 = 10 \cdot 10^6 \text{ N/m}$

En la siguiente gráfica se mostrará el desplazamiento horizontal del nodo tres con el tiempo, cuando la constante de rigidez del muelle se aleja del valor calculado en (4.20).



Gráfica 4.7 : Desplazamiento horizontal del nodo tres de la estructura uno frente al tiempo, aplicándose carga dinámica para muelles de rigidez:  $K_{\text{muelle1}}$ ,  $K_{\text{muelle2}}$  y  $K_{\text{muelle3}}$ .

El desplazamiento estático sería aproximadamente el que se produce para el muelle tres. Se puede observar que a menor constante de rigidez, mayor es el desplazamiento horizontal.

#### 4.1.4.2. Introducción de un amortiguador en la estructura

En este caso se planteará el problema en las mismas condiciones que el caso 4.1.4.1, solo que se añadirá un amortiguador en paralelo con el muelle.

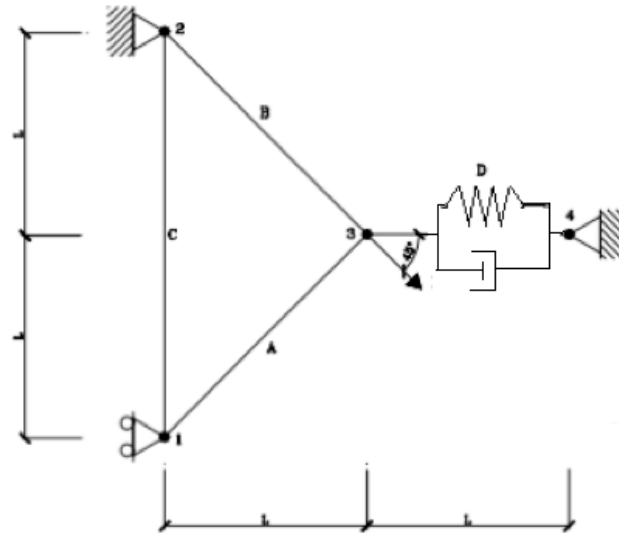


Figura 4.4: Estructura 1 con muelle y amortiguador.

Para verificar que los resultados obtenidos en el programa cuando se introducen amortiguadores en la estructura son correctos, se someterá a la estructura a una carga dinámica excitada con una de las frecuencias propias, en este caso  $\omega_3$ , cuyo valor se muestra en la tabla 4.9, se mantendrá el mismo valor de la constante de rigidez del muelle ( $K=10 \cdot 10^6$  N/m), y se irá variando el valor de la constante de amortiguamiento.

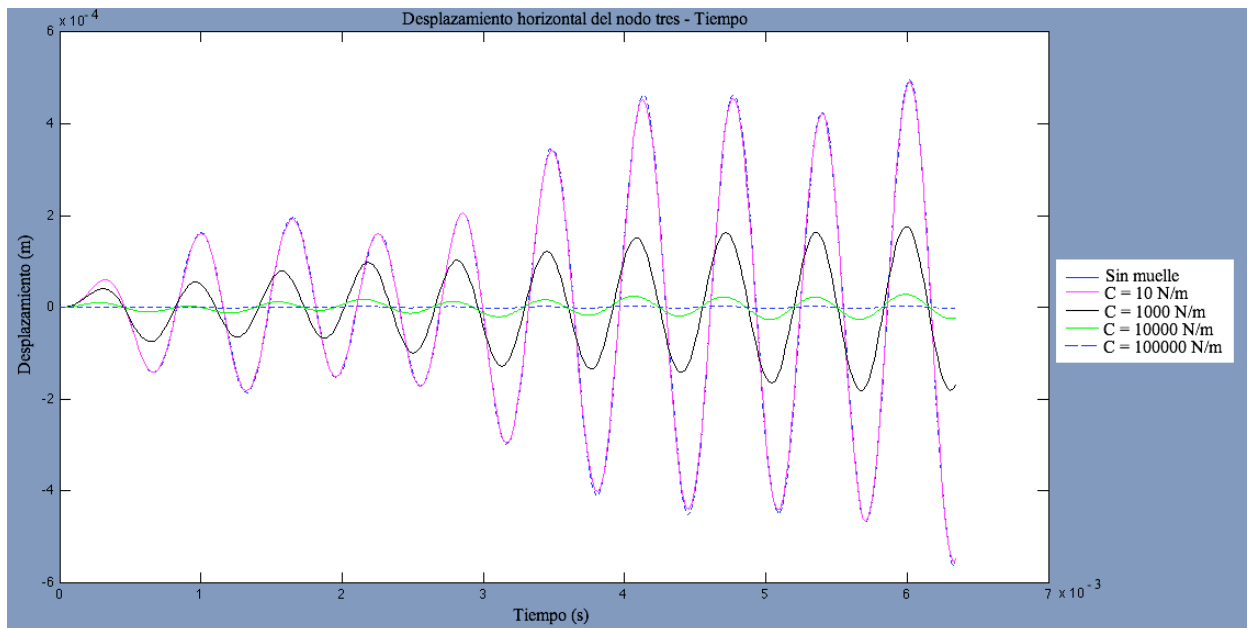
	Frecuencia propia (Hz)	Desplazamiento vertical del nodo uno	Desplazamiento horizontal del nodo tres	Desplazamiento horizontal del nodo tres	
$\omega_1$	4621,9	0,607	0,197	0,769	Modos propios
$\omega_2$	7637,5	0,609	0,448	-0,654	
$\omega_3$	9901,7	0,433	-0,888	-0,156	

Tabla 4.10: Frecuencias y modos propios de la estructura uno, con muelle ( $K=10 \cdot 10^6$  N/m) y amortiguador.

Los valores de la constante de amortiguamiento serán:

- a) Amortiguador 1  $\rightarrow C_1 = 10$  N·s/m.
- b) Amortiguador 2  $\rightarrow C_2 = 1000$  N·s /m.
- c) Amortiguador 3  $\rightarrow C_3 = 10000$  N·s /m.
- d) Amortiguador 4  $\rightarrow C_4 = 100000$  N·s /m.

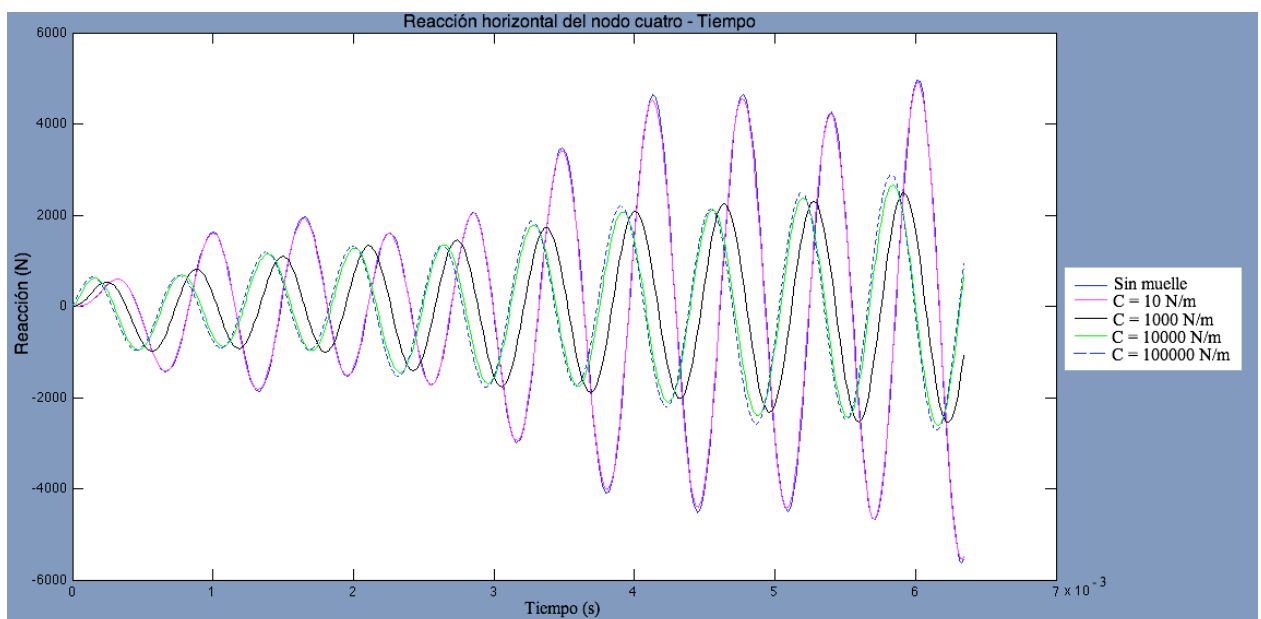
A continuación se muestra una gráfica en la que se representan los valores del desplazamiento horizontal del nodo tres frente al tiempo, aplicándose una carga dinámica excitada con la frecuencia propia  $\omega_3$ , para poder observar mejor como se reduce el valor del desplazamiento.



Gráfica 4.8 : Desplazamiento horizontal del nodo tres de la estructura uno frente al tiempo, aplicándose carga dinámica para muelle y amortiguadores:  $C_1$ ,  $C_2$ ,  $C_3$  y  $C_4$ .

A mayor constante de amortiguamiento, menor es el desplazamiento horizontal del nodo tres, consiguiéndose finalmente reducir la resonancia.

En la siguiente gráfica se mostrarán los valores de la reacción horizontal del nodo cuatro:



Gráfica 4.9: Reacción horizontal del nodo cuatro de la estructura uno frente al tiempo, aplicándose carga dinámica para muelle y amortiguadores:  $C_1$ ,  $C_2$ ,  $C_3$  y  $C_4$ .

El valor de la reacción disminuye para valores de constante de amortiguamiento de mayores.

## 4.2. Estructura 2

La siguiente estructura articulada tiene la misma sección en todas las barras aluminio, longitud de barras horizontales, verticales y distancia entre barras E y B de 0,5 m. Las características del tipo de sección y la fuerza de excitación son las mismas que para la estructura uno, y se muestran en la tabla 4.1.

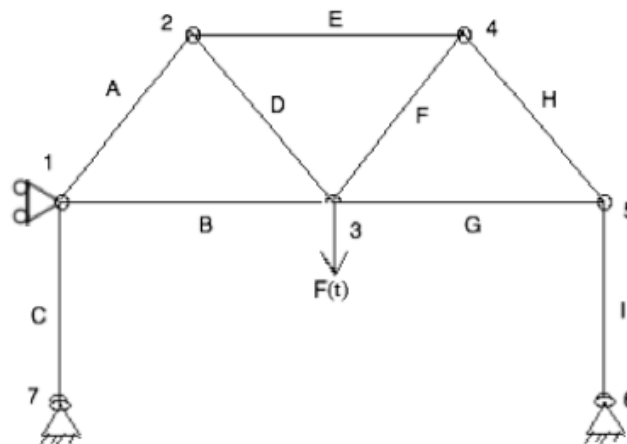


Figura 4.5: Estructura dos.

En primer lugar se obtendrá el valor de los desplazamientos en cada uno de los nodos, reacciones y fuerzas en Ed-Tridim, para comparar con los valores obtenidos en el caso dinámico.

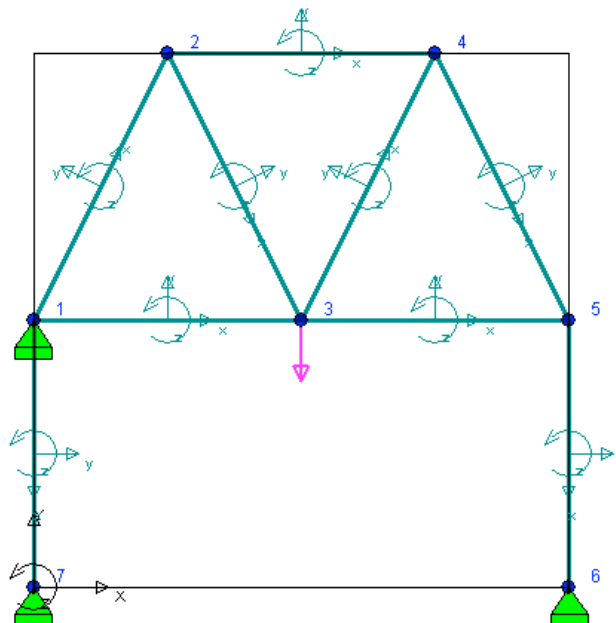


Figura 4.6: Estructura dos resuelta mediante Ed-Tridim.

En la siguiente tabla se muestran los resultados obtenidos mediante Ed-Tridim:

Nodo	Desplazamiento horizontal (m)	Desplazamiento vertical (m)	Reacción horizontal (N)	Reacción vertical (N)
1	0	$-4,55 \cdot 10^{-5}$	0	0
2	$4,55 \cdot 10^{-5}$	$-1,32 \cdot 10^{-4}$	0	0
3	$2,27 \cdot 10^{-5}$	$-2,01 \cdot 10^{-4}$	0	-1000
4	0	$-1,32 \cdot 10^{-4}$	0	0
5	$4,55 \cdot 10^{-5}$	$-4,55 \cdot 10^{-5}$	0	0
6	0	0	0	500
7	0	0	0	500

Tabla 4.11 : Tabla de resultados de desplazamiento y reacciones de la estructura dos, obtenidos mediante Ed-Tridim

### 4.2.1. Obtención de desplazamientos en dinámico

Se analizará el desplazamiento vertical del nodo tres y las reacciones del nodo seis, cuando se ejerza una fuerza dinámica cuya fuerza de excitación será de 1000 N y un valor de frecuencia angular menor a la frecuencia propia del sistema.

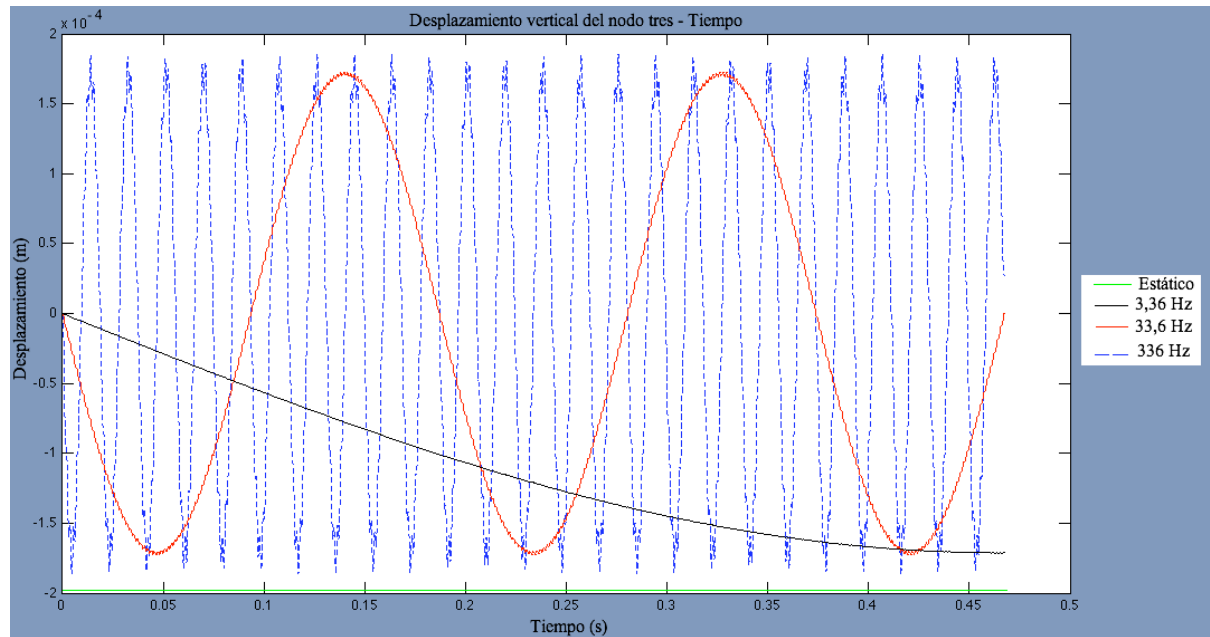
	Frecuencias propias (Hz)	Desplazamiento vertical del nodo tres	
$\omega_1$	$3,028 \cdot 10^3$	0,452	Modos propios
$\omega_2$	$3,360 \cdot 10^3$	0,595	
$\omega_3$	$5,593 \cdot 10^3$	0,133	
$\omega_4$	$7,788 \cdot 10^3$	-0,024	
$\omega_5$	$8,931 \cdot 10^3$	0,565	
$\omega_6$	$1,145 \cdot 10^4$	-0,062	
$\omega_7$	$1,282 \cdot 10^4$	-0,526	
$\omega_8$	$1,297 \cdot 10^4$	-0,192	
$\omega_9$	$1,382 \cdot 10^4$	-0,085	

Tabla 4.12: Frecuencias propias y valores de los modos propios relacionados con el desplazamiento vertical del nodo tres de la estructura dos.

Se puede observar en la tabla 4.10 que el mayor valor del vector de modos propios de desplazamiento vertical del nodo tres, se corresponde con la frecuencia  $\omega_2$ , por tanto, se analizará la estructura para las siguientes frecuencias:

- Frecuencia 1  $\rightarrow \omega = \omega_2/1000 = 3,36$  Hz.
- Frecuencia 2  $\rightarrow \omega = \omega_2/100 = 33,6$  Hz.
- Frecuencia 3  $\rightarrow \omega = \omega_2/10 = 336$  Hz.

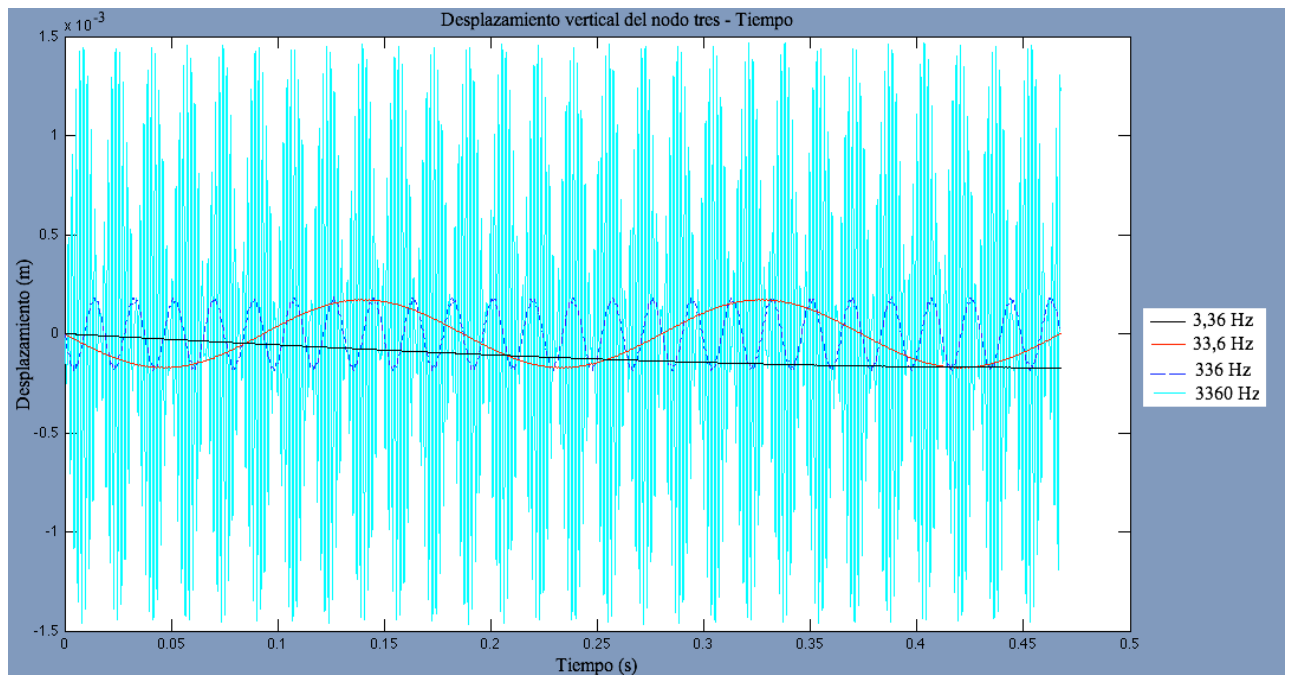




Gráfica 4.10 : Desplazamiento vertical del nodo tres de la estructura dos frente al tiempo, aplicándose carga dinámica para tres valores de frecuencia angular distintas.

Cuando la carga dinámica esta excitada con frecuencias angulares cercanas a la frecuencia propia de la estructura, los valores del desplazamiento son mayores. Por otro lado, se puede apreciar que el desplazamiento en el caso estático es mayor que en el caso dinámico, esto se debe a que en el caso estático no se considera la masa de las barras y por tanto se obtienen desplazamientos menores.

Si se comparan los desplazamientos calculados anteriormente con los que se produce para la frecuencia propia  $\omega_2$ , se podrá observar que los desplazamientos para frecuencias menores a la propia, son mucho menores.



Gráfica 4.11 : Desplazamiento vertical del nodo tres de la estructura dos frente al tiempo, aplicándose carga dinámica para tres valores de frecuencia angular distintas y la frecuencia propia de la estructura.

## 4.2.2. Introducción de muelles y amortiguadores en la estructura

### 4.2.2.1. Introducción de muelles en la estructura

En este apartado se sustituirá las barras C e I de la estructura, por muelles con distintas constantes de rigidez, y se someterá a la estructura a una carga dinámica, con una fuerza de excitación de 1000 N y una frecuencia angular  $\omega=33,6$  Hz.

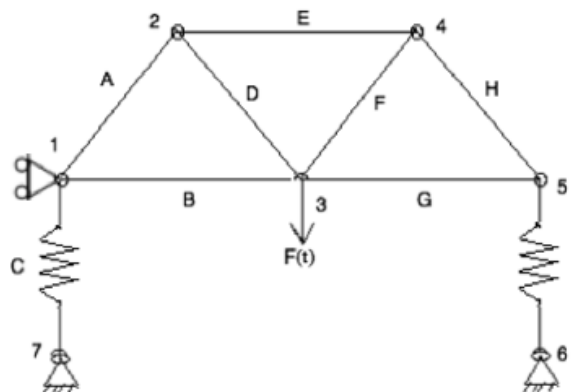


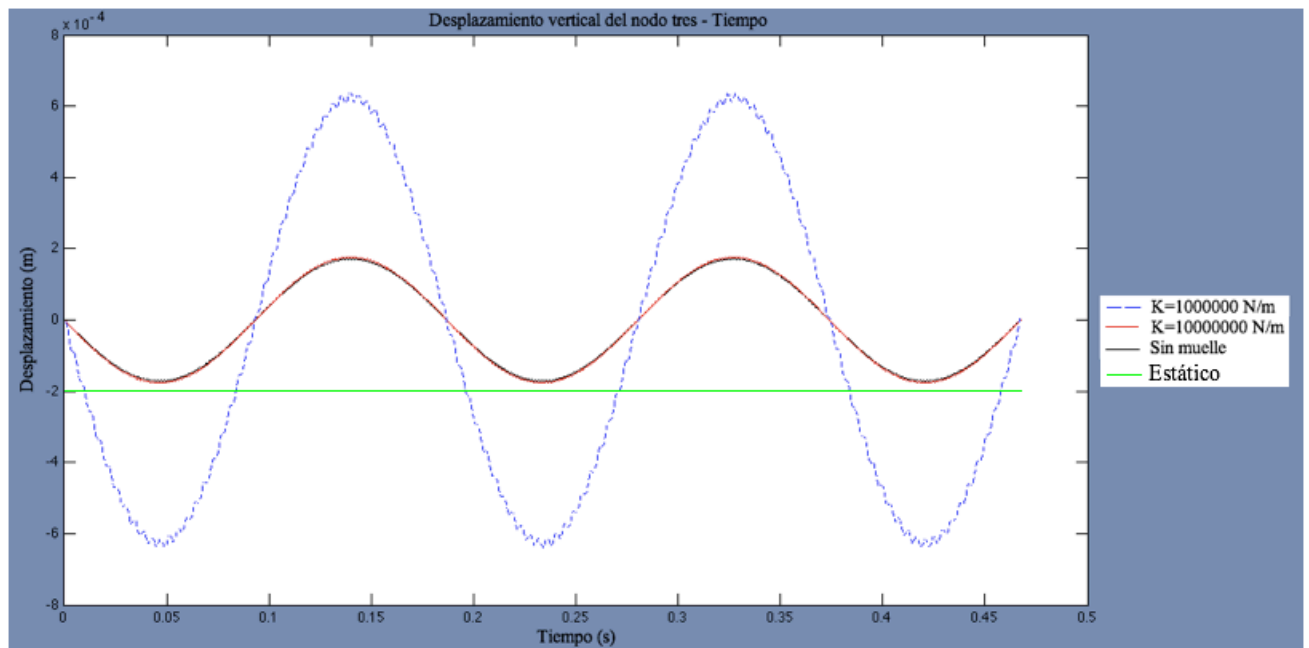
Figura 4.7: Estructura dos con muelles

Se analizará el desplazamiento vertical del nodo tres, cuando las barras C e I de la estructura no tengan muelles, y con muelles con una rigidez similar a la de las barras (4.20):

a) Muelle 1  $\rightarrow K=10 \cdot 10^6$  N/m

b) Muelle 2  $\rightarrow K=10 \cdot 10^5$  N/m

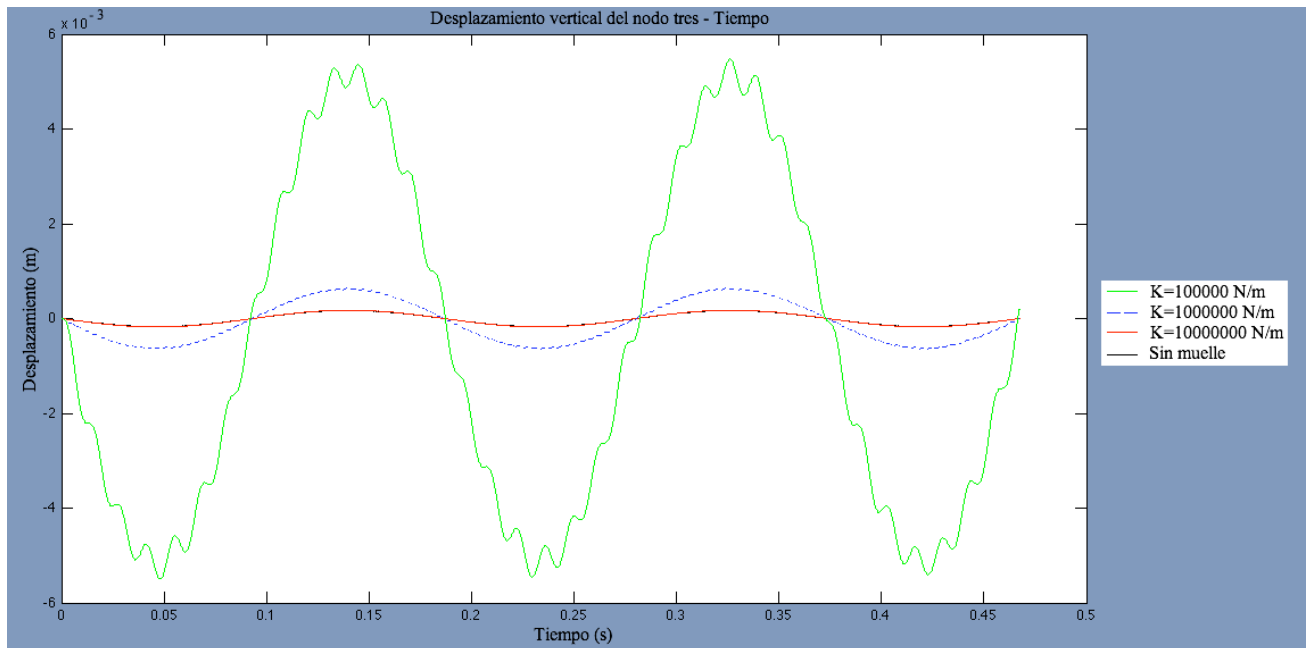
A continuación se muestra una gráfica en la que se observa el desplazamiento vertical del nodo tres para los distintos valores de constante de rigidez de los muelles.



Gráfica 4.12: Desplazamiento vertical del nodo tres de la estructura dos frente al tiempo, aplicándose carga dinámica cuando se varía el valor de la constante de rigidez de los muelles C e I.

Si los elementos C e I de la estructura, son, o bien barras, o bien muelles con una constante de rigidez similar a la de las barras, el desplazamiento es casi el mismo, ya que sus gráficas se superponen. Sin embargo, en el momento en que se reduce 10 veces la constante de rigidez de los muelles, el desplazamiento aumenta bastante.

En el siguiente caso, se observará como los desplazamientos mostrados en la gráfica 4.10 son casi insignificantes, comparados con el que se produce para un muelle  $K=10 \cdot 10^4$ .



Gráfica 4.13 : Desplazamiento vertical del nodo tres de la estructura dos frente al tiempo, aplicándose carga dinámica cuando se varía el valor de la constante de rigidez de los muelles C e I.

#### 4.2.2.2. Introducción de amortiguadores en la estructura

Para la siguiente estructura:

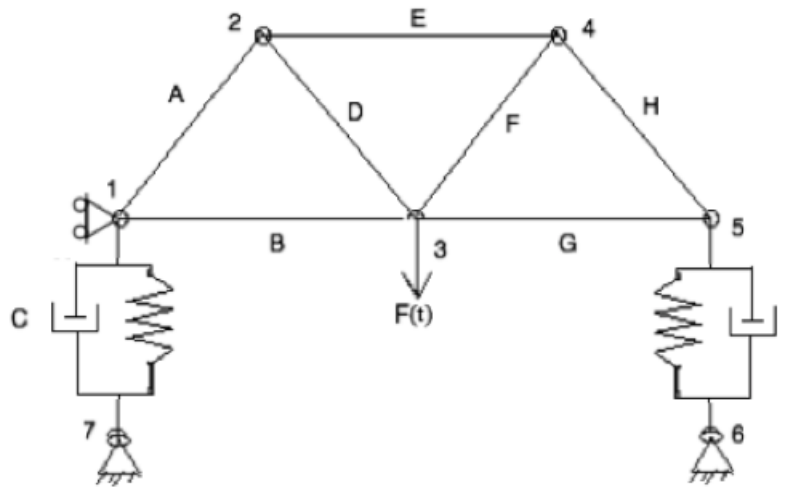


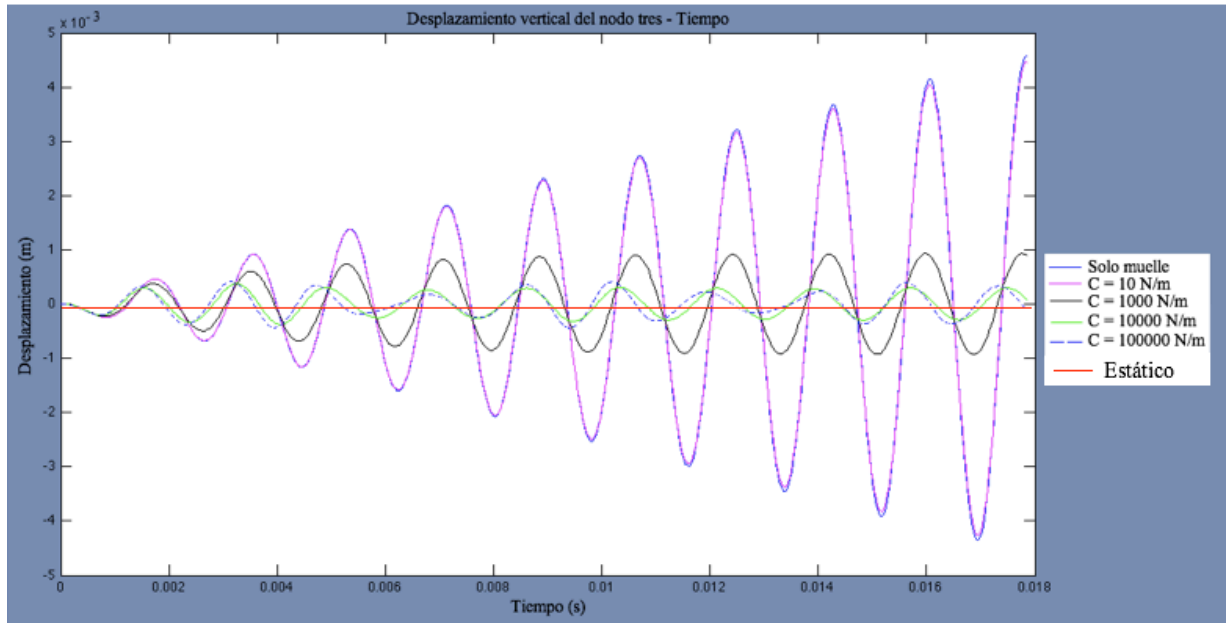
Figura 4.8: Estructura dos con muelles y amortiguadores.

Se analizarán los desplazamientos obtenidos, cuando se aplique una carga dinámica excitada con la frecuencia propia. Para ello habrá que recalcular los valores y modos propios de la estructura, puesto que al introducir muelles sus valores no serán los mismos.

	Frecuencias propias (Hz)	Desplazamiento vertical del nodo tres	
$\omega_1$	$2,998 \cdot 10^3$	0,463	Modos propios
$\omega_2$	$3,520 \cdot 10^3$	0,579	
$\omega_3$	$5,486 \cdot 10^3$	0,123	
$\omega_4$	$7,734 \cdot 10^3$	0,003	
$\omega_5$	$8,774 \cdot 10^3$	-0,571	
$\omega_6$	$1,138 \cdot 10^4$	0,062	
$\omega_7$	$1,276 \cdot 10^4$	0,463	
$\omega_8$	$1,293 \cdot 10^4$	-0,332	
$\omega_9$	$1,377 \cdot 10^4$	-0,083	

Tabla 4.13 : Frecuencias y modos propios para estructura dos con muelles.

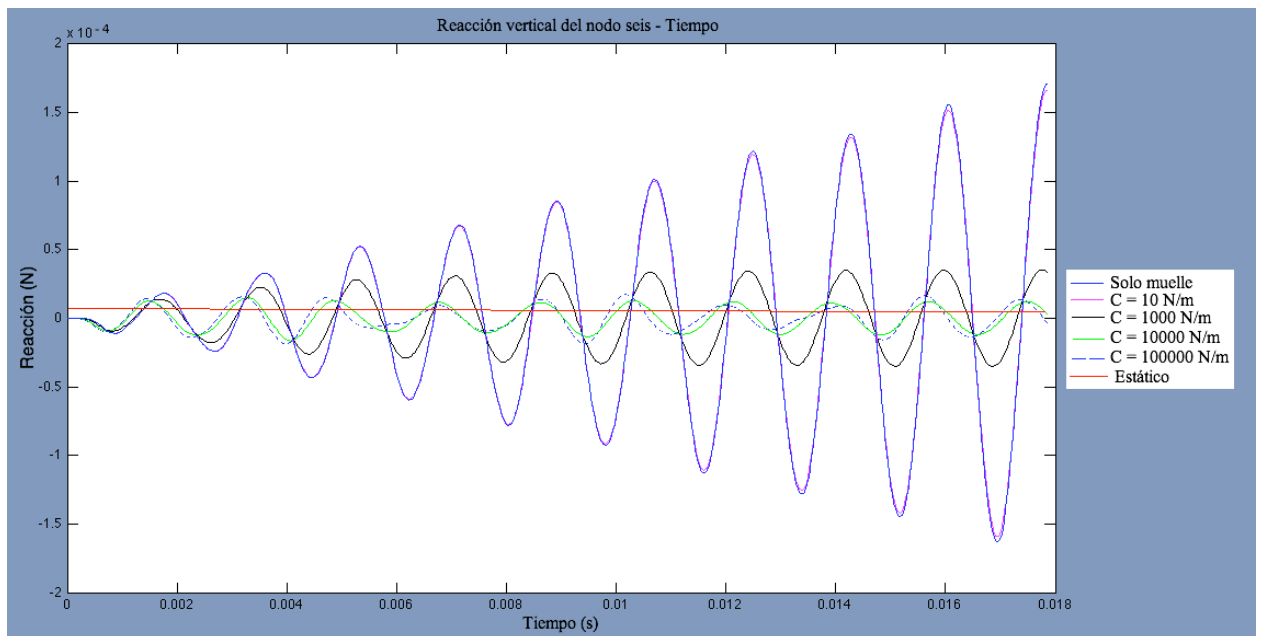
En la siguiente gráfica se mostrarán los resultados de desplazamientos verticales del nodo tres, cuando se aplique una carga dinámica con una frecuencia angular de 3520 Hz. Se mantendrá un valor constante de  $10 \cdot 10^6$  N/m de constante de rigidez del muelle y se cambiará de amortiguador para ver la reducción que se produce en el desplazamiento.



Gráfica 4.14: Desplazamiento vertical del nodo tres, para barras sustituidas por amortiguadores y muelles.

Aumentando el valor de la constante de amortiguamiento del amortiguador, se reduce el desplazamiento y a partir de valores superiores a 10000 N·s/m, la reducción de desplazamiento es casi la misma.

A continuación se puede observar el valor de las reacciones para los distintos amortiguadores:



Gráfica 4.15: Reacción vertical del nodo seis, para barras sustituidas por amortiguadores y muelles.

Aun obteniéndose valores altos de reacción vertical para la frecuencia propia de la estructura, ésta se reduce a medida que se aumenta el valor de la constante de amortiguamiento del amortiguador.

### 4.3. Estructura 3

El pórtico de la figura, esta formada por barras de aluminio de sección circular maciza y tiene aplicadas las siguientes cargas de viento.

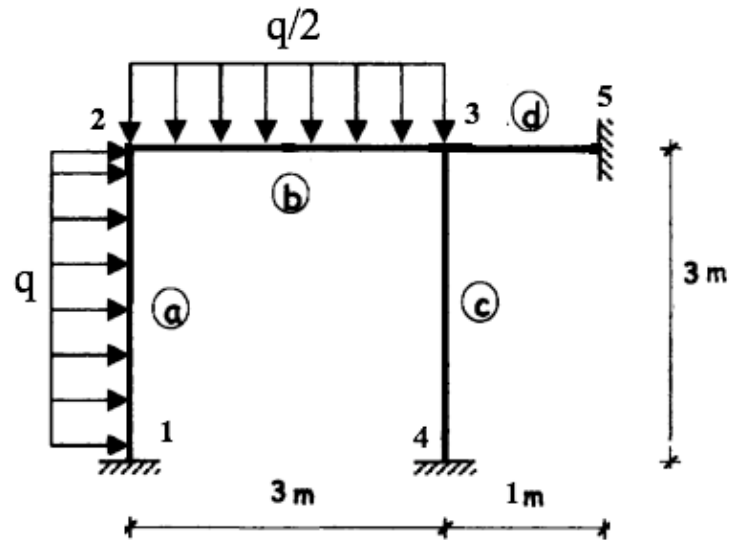


Figura 4.9 : Estructura tres.

Los datos para la resolución del problema son los siguientes:

Densidad (Kg/m <sup>3</sup> )	Modulo de elasticidad (N/m <sup>2</sup> )	Carga de viento (N/m)	Diámetro de la sección (m)
2700	$70 \cdot 10^9$	$1000/3$	0,01

Tabla 4.14: Datos estructura tres.

A partir de estos datos se obtiene el momento de inercia de las barras.

$$I = \pi \cdot R^4/2 = 9,817 \cdot 10^{-10} \text{ m}^4 \quad (4.21)$$

En primer lugar se calculará el valor de los desplazamientos, reacciones y fuerzas en cada uno de los nodos con Ed-Tridim, para comparar con los valores obtenidos en el caso dinámico.

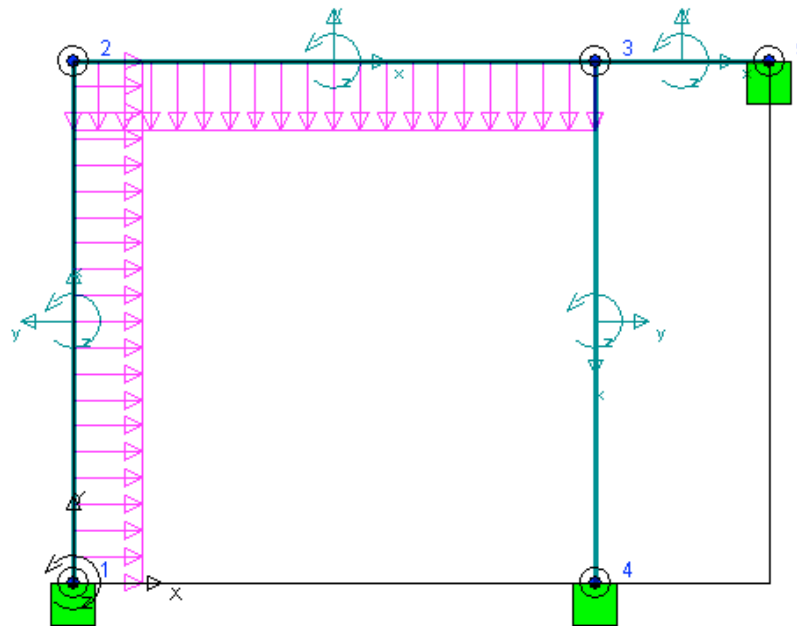


Figura 4.10 : Resolución con Ed-Tridim

En la siguiente tabla se muestran los resultados obtenidos:

Nodo	1	2	3	4	5
Desplazamiento horizontal (m)	0	$3,64 \cdot 10^{-4}$	$9,09 \cdot 10^{-5}$	0	0
Desplazamiento vertical (m)	0	$-1,36 \cdot 10^{-4}$	$-1,36 \cdot 10^{-4}$	0	0
Giro (rad)	0	0	0	0	0
Reacción horizontal (N)	-500	0	0	0	-500
Reacción vertical (N)	250	0	0	250	0
Momento (Nm)	0	0	0	0	0

Tabla 4.15 : Tabla de resultados de desplazamiento y reacciones de la estructura dos, obtenidos mediante Ed-Tridim



### 4.3.1. Obtención de desplazamientos y reacciones en dinámico

Se obtendrán los desplazamientos horizontales del nodo dos bajo carga dinámica, con distintos valores de frecuencia angular. Para ello, primero habrá que obtener las frecuencias propias de la estructura.

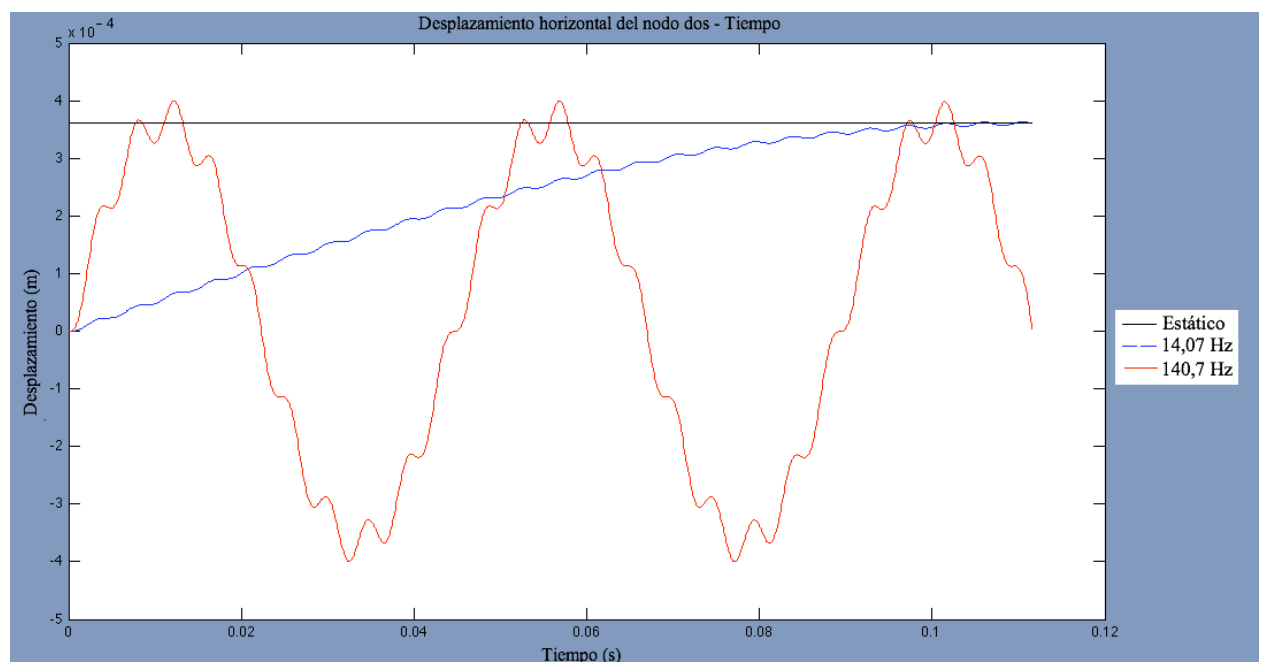
	Frecuencias propias (Hz)	Desplazamiento horizontal del nodo dos	
$\omega_1$	6,785	$-5,254 \cdot 10^{-6}$	Modos propios
$\omega_2$	10,945	$-2,246 \cdot 10^{-6}$	
$\omega_3$	$1,407 \cdot 10^3$	0,939	
$\omega_4$	$1,572 \cdot 10^3$	$8,828 \cdot 10^{-10}$	
$\omega_5$	$1,697 \cdot 10^3$	$-3,973 \cdot 10^{-10}$	
$\omega_6$	$3,283 \cdot 10^3$	-0,298	

Tabla 4.16: Frecuencias propias y valores de los modos propios relacionados con el desplazamiento horizontal del nodo dos de la estructura tres.

Se puede observar que la frecuencia propia que mayor desplazamiento horizontal produciría en el nodo dos sería  $\omega_3$ , por tanto, se analizará la estructura con las siguientes frecuencias angulares:

- a) Frecuencia 1  $\rightarrow \omega = \omega_3 / 100 = 14,07$  Hz.
- b) Frecuencia 2  $\rightarrow \omega = \omega_3 / 10 = 140,7$  Hz.

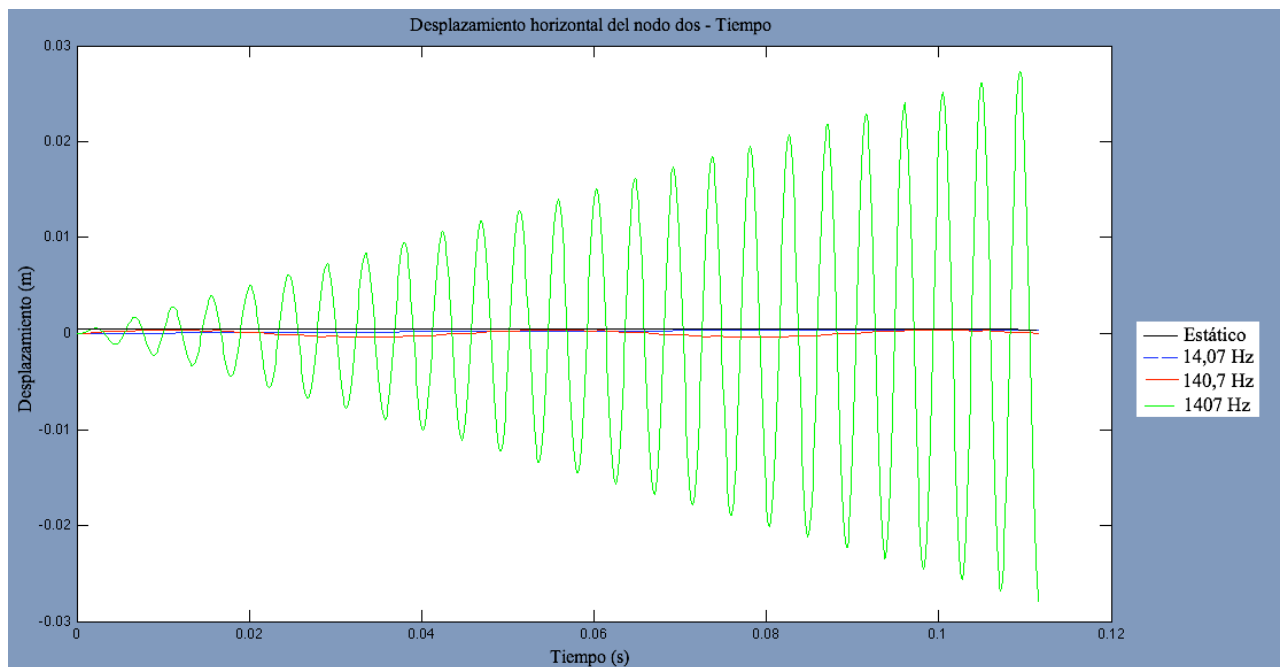
En la siguiente gráfica se muestran los resultados de los desplazamientos dinámicos para el caso estático, la frecuencia uno y dos.



Gráfica 4.16 :Desplazamiento horizontal del nodo dos para dos frecuencias angulares distintas y caso estático.

El valor de desplazamiento obtenido para la frecuencia de menor valor en un pico máximo, es muy parecida a la que se obtiene en el caso estático. Sin embargo, con una frecuencia mayor, mas cercana al valor de la frecuencia propia, los valores de desplazamiento obtenidos ya son un poco mayores.

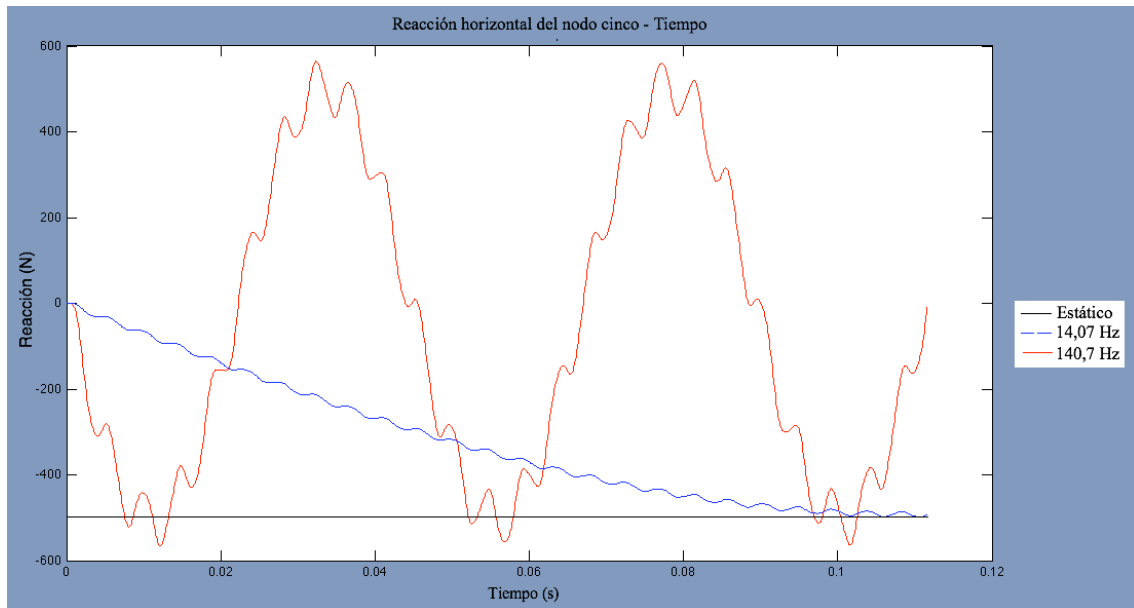
En la siguiente gráfica se estudiarán los desplazamientos introduciendo también la frecuencia propia:



Gráfica 4.17 :Desplazamiento horizontal del nodo dos para dos frecuencias angulares distintas, la frecuencia propia de la estructura y caso estático.

Se puede observar como aumenta el valor de los desplazamientos para la frecuencia propia, siendo los desplazamientos para las frecuencias angulares de la gráfica 4.16 insignificantes frente a ésta.

En la siguiente gráfica se muestran los resultados de la reacción horizontal del nodo cinco, para caso estático y dos valores de frecuencia angular.



Gráfica 4.18 :Reacción horizontal del nodo cinco para dos frecuencias angulares distintas y caso estático.

Para frecuencias angulares muy bajas, el resultado obtenido es similar al caso estático. A medida que se aumenta el valor de la frecuencia angular, aumenta el valor de las reacciones.

### 4.3.2. Introducción de muelle y amortiguador en la estructura

Se sustituirá la barra D por distintos amortiguadores, y un muelle de constante de rigidez  $K = 10 \cdot 10^6$  N/m, para reducir el valor del desplazamiento y la reacción.

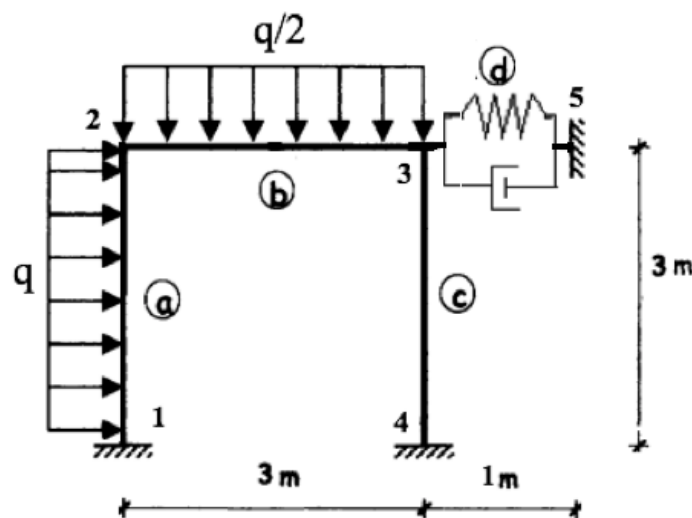


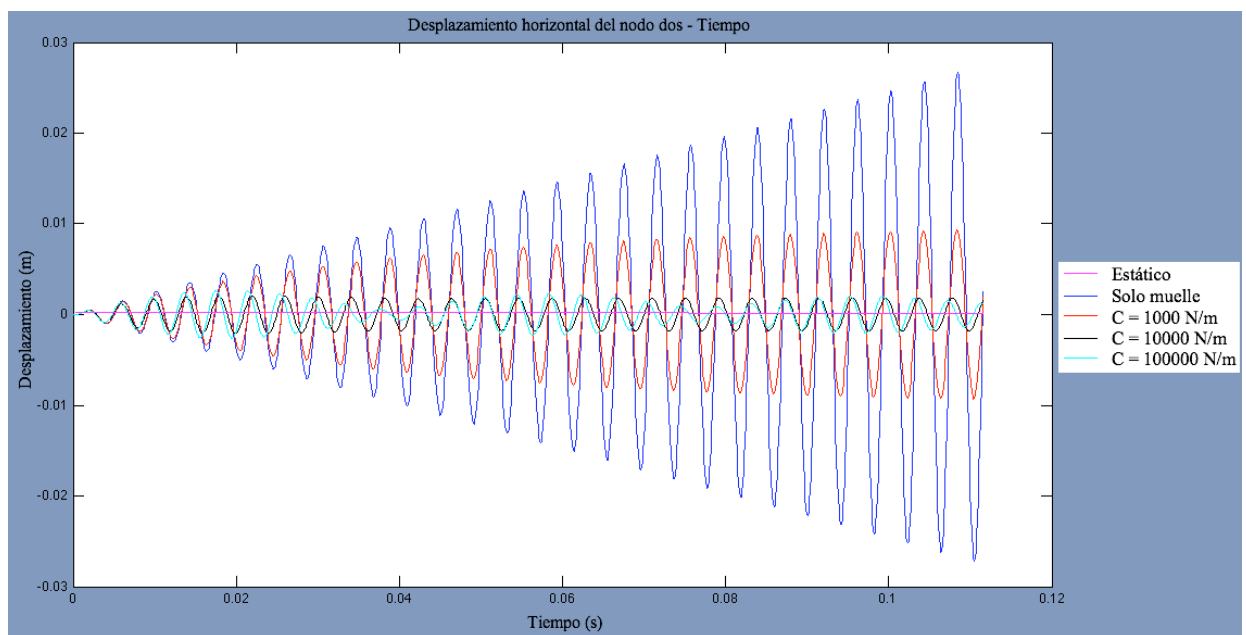
Figura 4.11 : Estructura tres con muelle y amortiguador.

Para analizar mejor el efecto del amortiguador, se excitará la carga dinámica con el valor de la frecuencia propia de la estructura. Por tanto, puesto que se ha eliminado una barra, habrá que recalcular las frecuencias y modos propios:

	Frecuencias propias (Hz)	Desplazamiento horizontal del nodo dos	
$\omega_1$	5,973	$-2,863 \cdot 10^{-6}$	Modos propios
$\omega_2$	7,712	$4,056 \cdot 10^{-6}$	
$\omega_3$	$1,535 \cdot 10^3$	0,978	
$\omega_4$	$1,571 \cdot 10^3$	$-3,042 \cdot 10^{-9}$	
$\omega_5$	$1,697 \cdot 10^3$	$6,533 \cdot 10^{-10}$	
$\omega_6$	$4,058 \cdot 10^3$	-0,179	

Tabla 4.17: Frecuencias propias y valores de los modos propios relacionados con el desplazamiento horizontal del nodo dos de la estructura tres.

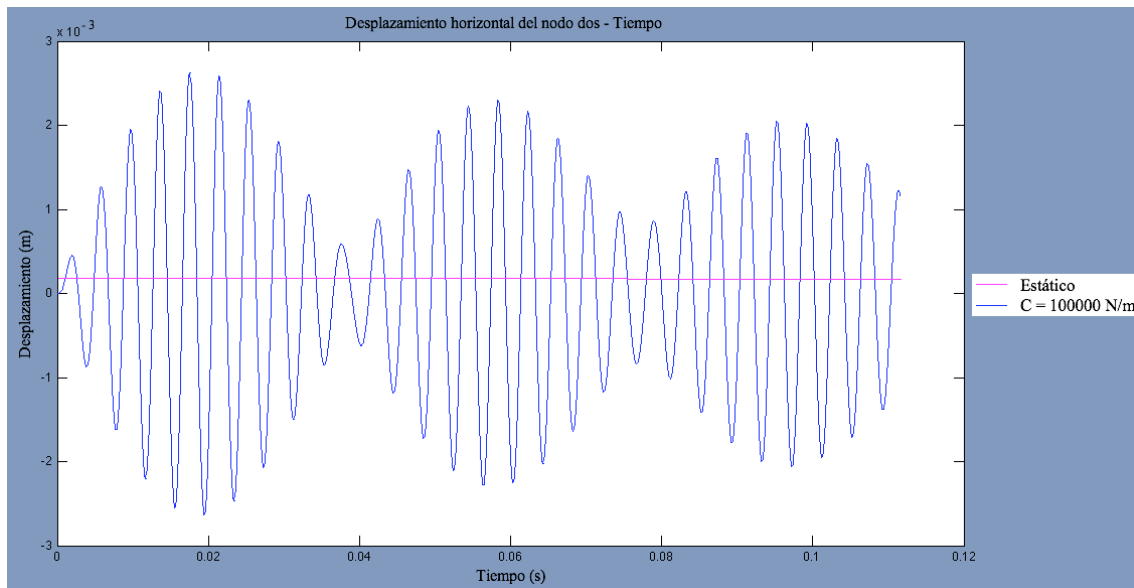
Se puede observar en la tabla 4.17, que a  $\omega_3$ , le corresponde el mayor desplazamiento, por tanto, en la siguiente gráfica se analizará el desplazamiento para distintos amortiguadores:



Gráfica 4.19 :Desplazamientos horizontales del nodo dos para la frecuencia propia , con distintos amortiguadores.

El valor de desplazamiento se reduce cuando la constante de amortiguamiento aumenta. Consiguiéndose para amortiguadores con constante de amortiguamiento superior a 1000 N·s/m, una reducción de la resonancia.

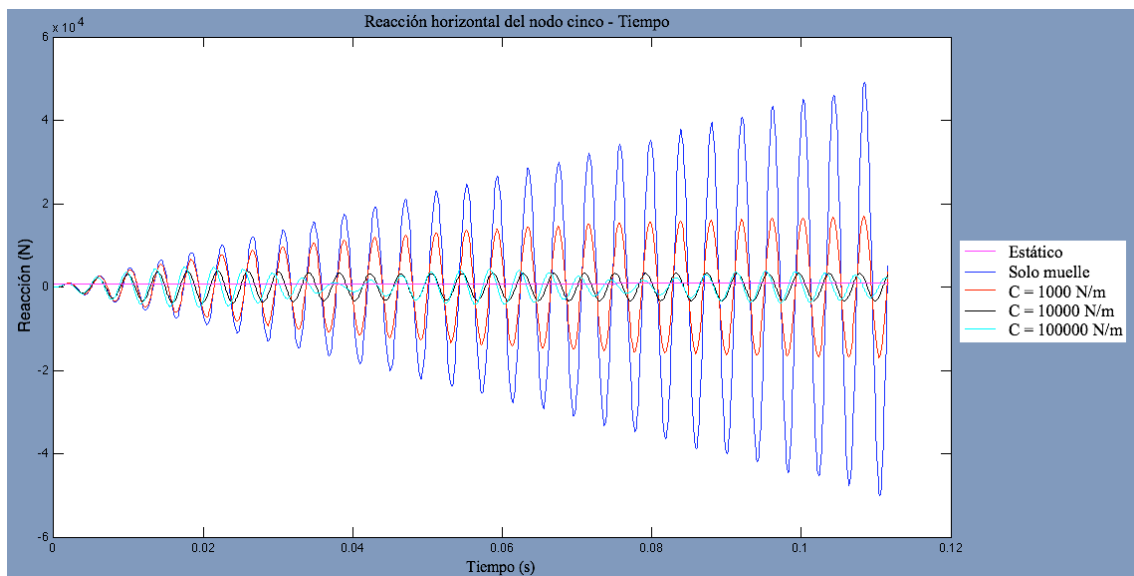
En la siguiente gráfica se puede ver el desplazamiento que se produce para una muelle de constante de amortiguamiento  $C = 100000 \text{ N}\cdot\text{s/m}$ , comparado con el caso estático.



Gráfica 4.20 :Desplazamientos horizontales del nodo dos para la frecuencia propia , con amortiguador  $C = 100000 \text{ N}\cdot\text{s/m}$ .

A medida que avanza el tiempo, el amortiguador reduce el valor del desplazamiento para una carga dinámica excitada con la frecuencia propia. Aun así, el desplazamiento es muy grande comparado con el caso estático.

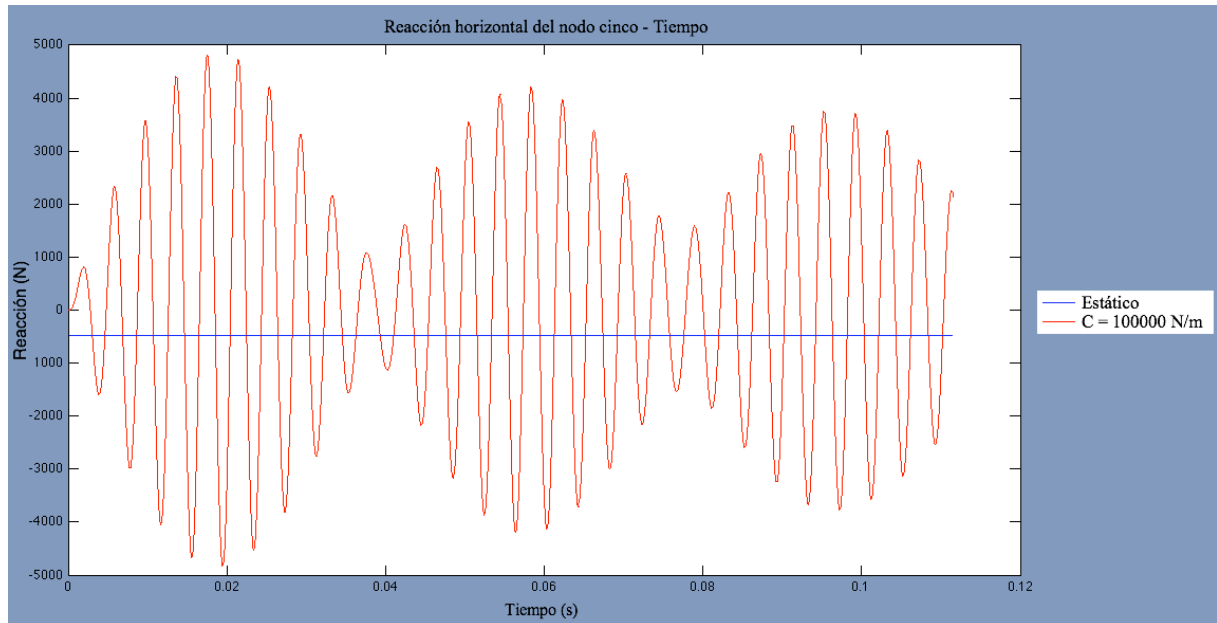
En la siguiente gráfica se tiene el resultado de las reacción horizontal del nodo 5, cuando se aplica una carga dinámica con frecuencia propia, y se varía el amortiguador.



Gráfica 4.21 :Reacción horizontal del nodo cinco para dos frecuencias angulares distintas y caso estático.

Como en el caso de desplazamiento, se reduce el valor de la reacción horizontal con constantes de amortiguamiento superiores a 1000 N·s/m.

Los resultados de la reacción horizontal para un amortiguador de constante de amortiguamiento  $C = 100000$  N·s/m, comparado con el caso estático.



Gráfica 4.22 :Reacción horizontal del nodo cinco para dos frecuencias angulares distintas y caso estático.

Como en el caso de los desplazamientos, a pesar de obtenerse valores de reacción horizontal muy elevadas, se puede observar como el amortiguador las va reduciendo progresivamente.

## 5. Conclusiones y trabajos futuros

## 5.1. Conclusiones

En este proyecto se ha llevado a cabo la realización del programa informático UCMM con su correspondiente interfaz, para poderse calcular desplazamientos, fuerzas y reacciones en cualquier tipo de estructura, ya sea articulada o reticulada.

La interfaz del programa UCMM ha sido desarrollada mediante MatLab, con el objetivo de que el usuario, de forma muy intuitiva y sin conocer en profundidad los procedimientos de cálculo, pueda definir y obtener los resultados de cualquier estructura plana que se plantee resolver.

El programa ha sido validado mediante comparación de resultados obtenidos con cálculos analíticos y otros programas informáticos.

Las principales características del programa desarrollado son:

- Cálculo dinámico simplificado.
- Utilización fácil e intuitiva del interfaz de usuario. Exponiendo el menú principal y demás ventanas de la interfaz de forma esquemática, de forma que el usuario pueda, en primer lugar definir el problema, y comprobar los resultados obtenidos posteriormente.
- Posibilidad de introducción de elementos disipativos en la estructura.
- Visualización de resultados de una forma sencilla, comprensible y didáctica. Mediante gráficas en las que se pueden visualizar y comparar distintos valores, y tablas en las que se explica cuál es el valor obtenido.
- Optimización de los cálculos.
- Minimizar las líneas de programación para conseguir un menor tiempo de resolución y visualización de resultados.
- No redundancia de introducción de datos.
- Cálculo y representación en tablas de frecuencias y modos propios de las estructuras que se planteen, lo que permite una mayor comprensión por parte del usuario del efecto de la resonancia.
- Posibilidad de modificación de la estructura.



## 5.2. Trabajos futuros

Este proyecto será la base un proyecto mas amplio, que tendrá como finalidad la realización de los siguientes objetivos:

- Cálculo de tensiones en barras de la estructura.
- Adición de masas puntuales en la estructura.
- Resolución de estructuras en tres dimensiones.
- Aplicación de fuerzas en barras, sin necesidad de descomponerlas en sus nodos.
- Posibilidad de introducción de elementos con comportamiento viscoelástico y elastoplástico.
- Posibilidad de introducción de distintos tipos de fuerzas dinámicas aplicadas.
- Validación del programa en condiciones dinámicas.

Además, se pueden realizar mejoras en la interfaz:

- Convertir UCMM en un programa ejecutable \*.exe, que permita su utilización sin necesidad de tener instalado el MatLab.
- Mejora de la programación de UCMM.
- Posibilidad de guardar e imprimir resultados obtenidos.
- Posibilidad de ejecutar este programa directamente desde internet.

## 6. Bibliografía

## Bibliografía

- [1] <http://ing.unne.edu.ar/pub> Pagina de Ingenieros de la Universidad Nacional del Nordeste.
- [2] [http://www.uhu.es/josemiguel.davila/TeoriaEstructuras\\_archivos/TeoriaEstructuras\\_TEMAVIII-3\\_Matricial-Ejemplo2.pdf](http://www.uhu.es/josemiguel.davila/TeoriaEstructuras_archivos/TeoriaEstructuras_TEMAVIII-3_Matricial-Ejemplo2.pdf) . José Miguel Dávila. Universidad de Huelva. Departamento de Ingeniería Minera, Mecánica y Energética.
- [3] <http://cypecad.cype.es/> Pagina web oficial del código comercial CYPE Ingenieros S.A.
- [4] David Fernández. “Práctica N°1: Estructuras Articuladas”. Universidad Carlos III de Madrid. Departamento de Mecánica de Medios Continuos y Teoría de Estructuras.
- [5] Javier Atencia, Raúl Nestar. (2001) “Aprenda MatLab 6.0 como si estuviera en primero”. Campus Tecnológico de la Universidad de Navarra.
- [6] Daniel de León Cardenas, Jorge Espinosa Caballero, José Luis Vargas Cruz. (2002) “Interfaces Gráficas en Matlab usando GUIDE”. Universidad Autónoma de Baja California. Unidad Tijuana.
- [7] Javier García Galón. (2007) “Aprenda MatLab 7.0 como si estuviera en primero”. Universidad Politécnica de Madrid Ediciones.

# 7. Anexos

## Anexo 1

En este anexo se encuentran los ficheros con los que se obtiene, tanto para estructuras articuladas como reticuladas, las matrices de rigidez de una barra y de un muelle, así como las matrices de amortiguamiento de un amortiguador en los ejes globales de la estructura.

### Matriz de rigidez de barras en ejes globales para estructuras articuladas.

```
function k=kglobalesarticuladasb(E,A,L,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa));(sin(alfa)) (cos(alfa))];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa));-(sin(alfa)) (cos(alfa))];
%Obtenci n de cada una de las submatrices de una barra en ejes locales
kl11=[(E*A)/L 0;0 0];
kl12=[-(E*A)/L 0;0 0];
kl21=[-(E*A)/L 0;0 0];
kl22=[(E*A)/L 0;0 0];
%Obtenci n de cada una de las submatrices de una barra en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de rigidez de una barra en ejes globales
k=[kg11 kg12;kg21 kg22];
```

### Matriz de rigidez de muelles en ejes globales para estructuras articuladas.

```
function k=kglobalesmuellearticuladas(kmuelle,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa));(sin(alfa)) (cos(alfa))];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa));-(sin(alfa)) (cos(alfa))];
%Obtenci n de cada una de las submatrices de un muelle en ejes locales
kl11=[kmuelle 0;0 0];
kl12=[kmuelle 0;0 0];
kl21=[kmuelle 0;0 0];
kl22=[kmuelle 0;0 0];
%Obtenci n de cada una de las submatrices de un muelle en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de rigidez de un muelle en ejes globales
k=[kg11 kg12;kg21 kg22];
```

### Matriz de amortiguamiento en ejes globales para estructuras articuladas.

```
function k=kglobalesamortiguadorarticuladas(camortiguador,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa));(sin(alfa)) (cos(alfa))];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa));-(sin(alfa)) (cos(alfa))];
%Obtenci n de cada una de las submatrices de un amortiguador en ejes
locales
kl11=[camortiguador 0;0 0];
```

```
kl12=[camortiguador 0;0 0];
kl21=[camortiguador 0;0 0];
kl22=[camortiguador 0;0 0];
%Obtenci n de cada una de las submatrices de un amortiguador en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de amortiguamiento de un amortiguador en ejes globales
k=[kg11 kg12;kg21 kg22];
```

### Matriz de rigidez de barras en ejes globales para estructuras reticuladas.

```
function k=kglobalesreticuladasb(E,A,L,I,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa)) 0;(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa)) 0;-(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Obtenci n de cada una de las submatrices de una barra en ejes locales
kl11=[((E*A)/L) 0 0;0 ((12*E*I)/(L^3)) ((6*E*I)/(L^2));0
((6*E*I)/(L^2)) ((4*E*I)/L)];
kl12=[((-E*A)/L) 0 0;0 ((-12*E*I)/(L^3)) ((6*E*I)/(L^2));0 ((-
6*E*I)/(L^2)) ((2*E*I)/L)];
kl21=[((-E*A)/L) 0 0;0 ((-12*E*I)/(L^3)) ((-6*E*I)/(L^2));0
((6*E*I)/(L^2)) ((2*E*I)/L)];
kl22=[((E*A)/L) 0 0;0 ((12*E*I)/(L^3)) ((-6*E*I)/(L^2));0 ((-
6*E*I)/(L^2)) ((4*E*I)/L)];
%Obtenci n de cada una de las submatrices de una barra en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de rigidez de una barra en ejes globales
k=[kg11 kg12;kg21 kg22];
```

### Matriz de rigidez de muelle en ejes globales para estructuras reticuladas.

```
function k=kglobalesmuellereticuladas(kmuelle,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa)) 0;(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa)) 0;-(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Obtenci n de cada una de las submatrices de un muelle en ejes locales
kl11=[kmuelle 0 0;0 0 0;0 0 0];
kl12=[kmuelle 0 0;0 0 0;0 0 0];
kl21=[kmuelle 0 0;0 0 0;0 0 0];
kl22=[kmuelle 0 0;0 0 0;0 0 0];
%Obtenci n de cada una de las submatrices de un muelle en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de rigidez de un muelle en ejes globales
k=[kg11 kg12;kg21 kg22];
```

### Matriz de amortiguamiento en ejes globales para estructuras reticuladas.

```
function k=kglobalesamortiguadorreticuladas(camortiguador,alfa)
%Matriz de giro
T=[(cos(alfa)) -(sin(alfa)) 0;(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Matriz de giro traspuesta
Tt=[(cos(alfa)) (sin(alfa)) 0;-(sin(alfa)) (cos(alfa)) 0;0 0 1];
%Obtenci n de cada una de las submatrices de un amortiguador en ejes
locales
kl11=[camortiguador 0 0;0 0 0;0 0 0];
kl12=[camortiguador 0 0;0 0 0;0 0 0];
kl21=[camortiguador 0 0;0 0 0;0 0 0];
kl22=[camortiguador 0 0;0 0 0;0 0 0];
%Obtenci n de cada una de las submatrices de un amortiguador en ejes
globales
kg11=(T*kl11*Tt);
kg12=(T*kl12*Tt);
kg21=(T*kl21*Tt);
kg22=(T*kl22*Tt);
%Matriz de amortiguamiento de un amortiguador en ejes globales
k=[kg11 kg12;kg21 kg22];
```

## Anexo 2

En este fichero se encuentran los ficheros en los que se realizan los c lculos previos en estructuras articuladas y reticuladas, para la obtenci n de datos necesarios para los c lculos posteriores de desplazamientos y fuerzas.

### C lculos previos para estructuras reticuladas

```
function
[O,H,despvelred,dimensionkestructura,despvel,M,K,C]=calculos_previos_d
esplazamientos_ER(E,A,I,densidad,BARRAS,Nodo,RestriccionesNodo,Barramu
elle,Barraamortiguador)

%TODO LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
%INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA

%Vector que contiene el valor del m dulo de elasticidad de cada barra

%E(1,:)=70e9;
%E(2,:)=70e9;
%E(3,:)=70e9;
%E(4,:)=70e9;
%E(5,:)=70e9;
%E(6,:)=70e9;
%E(7,:)=70e9;

%Vector que contiene el valor del  rea de cada barra

%A(1,:)=pi*(((1e-2)/2)^(2));
%A(2,:)=pi*(((1e-2)/2)^(2));
%A(3,:)=pi*(((1e-2)/2)^(2));
%A(4,:)=pi*(((1e-2)/2)^(2));
%A(5,:)=pi*(((1e-2)/2)^(2));
%A(6,:)=pi*(((1e-2)/2)^(2));
%A(7,:)=pi*(((1e-2)/2)^(2));

%vector que contiene el valor de los momentos de inercia de cada barra
```

```
%I(1,:)=pi*(((1e-2)/2)^(4))/2);
%I(2,:)=pi*(((1e-2)/2)^(4))/2);
%I(3,:)=pi*(((1e-2)/2)^(4))/2);
%I(4,:)=pi*(((1e-2)/2)^(4))/2);
%I(5,:)=pi*(((1e-2)/2)^(4))/2);
%I(6,:)=pi*(((1e-2)/2)^(4))/2);
%I(7,:)=pi*(((1e-2)/2)^(4))/2);

%Vector densidad que contiene el valor de la densidad de cada una de
las
%barras

%densidad(1,:)=2700;
%densidad(2,:)=2700;
%densidad(3,:)=2700;
%densidad(4,:)=2700;
%densidad(5,:)=2700;
%densidad(6,:)=2700;
%densidad(7,:)=2700;

%Aquí se introducen los nodos de cada de una de las barras

%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];

%En la variable SIZEBARRAS se guardar· un vector que indicar· la
dimensión de la matriz BARRAS

SIZEBARRAS=size(BARRAS);

%Aquí se crea una matriz con las coordenadas de los nodos, en la que
el
número de fila se corresponder· con el número del nodo. El primer
valor (primera columna)
%ser· la coordenada 'x' y el segundo (segunda columna) la coordenada
'y'.

%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];
dimensionkeestructura=size(Nodo);
%Aqui se introduce la union del nodo menor con el mayor formando una
matriz
%de tantas filas como barras tenga la estructura y cuyos dos primeros
%valores son las coordenadas del nodo menor, y las otras dos valores
son
%las coordenadas del nodo mayor
for i=1:SIZEBARRAS(1,1)
    Barra(i,:)= [Nodo((BARRAS(i,1))),:] Nodo((BARRAS(i,2))),:]);
end

barrassize=size (Barra);
%Aqui se creara un vector con tantas filas como barras
%tenemos, e indicaremos con el valor de la cte de rigidez del muelle
si es un barra-muelle y con un 0 si es
%una barra normal

%Barramuelle(1,:)= [0];
%Barramuelle(2,:)= [0];
%Barramuelle(3,:)= [0];
%Barramuelle(4,:)= [0];
%Barramuelle(5,:)= [0];
```



```
%Barramuelle(6,:)=0];
%Barramuelle(7,:)=0];

%aquí creamos una matriz columna en la que diremos si la barra tiene
un
%amortiguador, dando el valor de cte de amortiguacion
%Barraamortiguador(1,:)=0];
%Barraamortiguador(2,:)=0];
%Barraamortiguador(3,:)=0];
%Barraamortiguador(4,:)=0];
%Barraamortiguador(5,:)=0];
%Barraamortiguador(6,:)=0];
%Barraamortiguador(7,:)=0];

%matriz que nos dice las restricciones de cada nodo, tal que el primer
%valor es desplazamiento horizontal, el segundo es desplazamiento
vertical
%y el tercero es giro. Y la primera fila corresponde al primer nodo,
la
%segunda al segundo y así sucesivamente. El valor cero indica que no
se
%produce desplazamiento del tipo que sea en ese nodo, mientras que el
uno
%indica que ese grado de libertad no está restringido y que por tanto
si se
%puede producir desplazamiento
%CONTAR CUANTOS UNOS HAY PORQUE LE IMPONDEREMOS TANTAS CONDICIONES DE
%CONTORNO EN EL ODE45 COMO NUMEROS DE UNOS MULTIPLICADOS POR DOS
%TENGAMOS.

%RestriccionesNodo(1,:)=0 0 1];
%RestriccionesNodo(2,:)=1 1 1];
%RestriccionesNodo(3,:)=1 1 1];
%RestriccionesNodo(4,:)=1 1 1];
%RestriccionesNodo(5,:)=0 0 1];

%Bucle for e if que se calcula a partir de las coordenadas de los
nodos las
%longitudes de las barras, sus ángulos y las k de cada barra
agrupandolas
%en una matriz de 6 filas X n columnas

for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
    angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
```

```
        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==Barra(i,3)) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3)) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end

%Para obtener las matrices de rigides en ejes globales de cada una de
las
%barras y muelles

for i=1:(barrassize(1,1))
    if (i==1) && (Barramuelle(i,1)==0)

k=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alfa(i,1)));
        elseif (i==1) && (Barramuelle(i,1)~=0)
            k=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        elseif i>1 && (Barramuelle(i,1)==0)
            k(1:6,
                    [((2*i)*3)-
5:((2*i)*3)])=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alf
a(i,1)));
        elseif (i>1) && (Barramuelle(i,1)~=0)
            k(1:6,
                    [((2*i)*3)-
5:((2*i)*3)])=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        end

    end

%Para obtener las matrices de amortiguacion en ejes globales de cada
una de las barras

for i=1:(barrassize(1,1))
    if (i==1)

camortig=kglobalesamortiguadorreticuladas(Barraamortiguador(i,1),alfa(
i,1));
        elseif (i>1)
            camortig(1:6,
                    [((2*i)*3)-
5:((2*i)*3)])=kglobalesamortiguadorreticuladas(Barraamortiguador(i,1),
alfa(i,1));
        end

    end

%Bucle para crear las matriz de ceros, unos y menos unos de las
matrices
%que conforman el sistema de ecuaciones

dimensionkestructura=size(Nodo);
dimensionK=dimensionkestructura(1,1);
for i=1:dimensionK
    if i==1
        CEROS=zeros(3);
        UNOS=eye(3);
        UNOSN=-eye(3);
    elseif i>1
        %filas
```

```

        CEROS(((i*3)-2):(i*3), [1:(3*i)])=zeros(3,(i*3));
        %columnas
        CEROS(1:(i+(i-3)), [((3*i)-2):(3*i)])=zeros((i+(i-3)),3);
        %matrices de unos y unos negativos
        UNOS(((2*i)+(i-2)):(3*i)), [((2*i)+(i-2)):(3*i)]=eye(3);
        UNOSN(((2*i)+(i-2)):(3*i)), [((2*i)+(i-2)):(3*i)]=-
eye(3);
    end
end

```

%Bucle for e if que se calcula la matriz de rigidez en ejes globales

```

kglobales=CEROS;
for i=1:(barrassize(1,1))
    if (i==1)
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m,[ (2*m)+(m-2):3*m])+k(1:3, [1:3]);
        %m,n
        kglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m,[ (2*n)+(n-2):3*n])+k(1:3, [4:6]);
        %n,m
        kglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n,[ (2*m)+(m-2):3*m])+k(4:6, [1:3]);
        %n,n
        kglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n,[ (2*n)+(n-2):3*n])+k(4:6, [4:6]);

        elseif i>1
            m=BARRAS(i,1);
            n=BARRAS(i,2);
            %m,m
            kglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m,[ (2*m)+(m-2):3*m]))+(k(1:3, [ (4*i)+(i+(i-5)): (4*i)+(i+(i-
3))] ))+(kglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m]));
            %m,n
            kglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m,[ (2*n)+(n-2):3*n]))+(k(1:3, [ (5*i)+(i-
2):6*i] ))+(kglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n]));
            %n,m
            kglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n,[ (2*m)+(m-2):3*m]))+(k(4:6, [ (4*i)+(i+(i-5)): (4*i)+(i+(i-
3))] ))+(kglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m]));
            %n,n
            kglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n,[ (2*n)+(n-2):3*n]))+(k(4:6, [ (5*i)+(i-
2):6*i] ))+(kglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n]));

    end

end

K=kglobales;

```

%Bucle for e if que se calcula la matriz de rigidez en ejes globales

```

cglobales=CEROS;
for i=1:(barrassize(1,1))
    if (i==1)
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        cglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m,[ (2*m)+(m-2):3*m])+camortig(1:3,[1:3]);
        %m,n
        cglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m,[ (2*n)+(n-2):3*n])+camortig(1:3,[4:6]);
        %n,m
        cglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n,[ (2*m)+(m-2):3*m])+camortig(4:6,[1:3]);
        %n,n
        cglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n,[ (2*n)+(n-2):3*n])+camortig(4:6,[4:6]);

    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        cglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m,[ (2*m)+(m-2):3*m]))+(camortig(1:3,[ (4*i)+(i+(i-
5)):(4*i)+(i+(i-3))]))+(cglobales((2*m)+(m-2):3*m,[ (2*m)+(m-2):3*m]));
        %m,n
        cglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m,[ (2*n)+(n-2):3*n]))+(camortig(1:3,[ (5*i)+(i-
2):6*i]))+(cglobales((2*m)+(m-2):3*m,[ (2*n)+(n-2):3*n]));
        %n,m
        cglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n,[ (2*m)+(m-2):3*m]))+(camortig(4:6,[ (4*i)+(i+(i-
5)):(4*i)+(i+(i-3))]))+(cglobales((2*n)+(n-2):3*n,[ (2*m)+(m-2):3*m]));
        %n,n
        cglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n,[ (2*n)+(n-2):3*n]))+(camortig(4:6,[ (5*i)+(i-
2):6*i]))+(cglobales((2*n)+(n-2):3*n,[ (2*n)+(n-2):3*n]));

end

end

C=cglobales;

%Bucle for e if para ordenar en forma de columna las restricciones de
los
%nodos de forma que en despvel quedan recogidas las gdl de cada
%nodo y de donde podremos sacar la reducida a la que llamaremos
%despvelred

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif i==1
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==1)
            desplazamiento((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))*(RestriccionesNodo(i,I));

```

```

elseif (i>1) && (I==2)
    desplazamiento((i*I)+(i-1),1)=((i*I)+(i-
1))*(RestriccionesNodo(i,I));
elseif (i>1) && (I==3)

desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
end
end
end

sizedesplazamiento=size(desplazamiento);

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)

velocidades=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
elseif i==1

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i
,I);
elseif (i>1) && (I==1)
    velocidades((i*I)+(i+(i-2)),1)=(((i*I)+(i+(i-
2)))+sizedesplazamiento(1,1))*(RestriccionesNodo(i,I));
elseif (i>1) && (I==2)
    velocidades((i*I)+(i-1),1)=(((i*I)+(i-
1))+sizedesplazamiento(1,1))*(RestriccionesNodo(i,I));
elseif (i>1) && (I==3)

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*(RestriccionesNodo(
i,I));
end
end
end

%despvel aporta el vector columna total con los
%desplazamientos y las velocidades

despvel=[desplazamiento;velocidades];
sizedespvel=size(despvel);

%Bucle para obtener la posicion de la primera incognita de
desplazamientos
%y velocidades en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end
%Con esto se obtiene un vector columna con las posiciones de las
velocidades y desplazamientos que son incognita

dvr=despvelred;

for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end
end

```

```

sizedespvelred=size(despvelred);

%Obtencion de la matriz de masas

MASAS=zeros(3,dimensionkestructura(1,1)*3);

for i=1:(barrassize(1,1))
    mas=densidad(i,1)*A(i,1)*L(1,i);
    for I=1:2
        MASAS(1:3,
                2):3*BARRAS(i,I))=(MASAS(1:3,
                2):3*BARRAS(i,I)))+[ (mas/2)      0      0;0      (mas/2)      0;0      0      0
        ((mas*((L(1,i))^(2)))/3)];
    end
end

for i=1:dimensionK
    if i==1
        M=MASAS(1:3, [1:3]);
    elseif i>1
        M(((i*3)-2):(i*3), [((i*3)-2):(i*3)])=MASAS(1:3, [((i*3)-2):(i*3)]);
    end
end

%Matrices que plantean el sistema de ecuaciones

h=[CEROS UNOSN;K C];
g=[UNOS CEROS;CEROS M];

%Para la obtencion de las matrices reducidas que contienen a la matriz
de
%rigidez y a la matriz de masas se emplea el siguiente bucle

for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[h(despvelred(i,1),despvelred(I,1))];
            G=[g(despvelred(i,1),despvelred(I,1))];
        elseif i>0
            H(i,I)=h(despvelred(i,1),despvelred(I,1));
            G(i,I)=g(despvelred(i,1),despvelred(I,1));
        end
    end
end

%Matriz inversa de G

O=inv(G);

```

### Cálculos previos para estructuras articuladas

```

function
[O,H,despvelred,dimensionkestructura,despvel,M,K,C]=calculos_previos_d
esplazamientos_EA(E,A,densidad,BARRAS,Nodo,RestriccionesNodo,Barramuel
le,Barraamortiguador)
%TODO LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
%INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA

%Vector que contiene el valor del módulo de elasticidad de cada barra

```

```
%E(1,:)=70e9;
%E(2,:)=70e9;
%E(3,:)=70e9;
%E(4,:)=70e9;
%E(5,:)=70e9;
%E(6,:)=70e9;
%E(7,:)=70e9;

%Vector que contiene el valor del área de cada barra

%A(1,:)=pi*(((1e-2)/2)^(2));
%A(2,:)=pi*(((1e-2)/2)^(2));
%A(3,:)=pi*(((1e-2)/2)^(2));
%A(4,:)=pi*(((1e-2)/2)^(2));
%A(5,:)=pi*(((1e-2)/2)^(2));
%A(6,:)=pi*(((1e-2)/2)^(2));
%A(7,:)=pi*(((1e-2)/2)^(2));

%Vector densidad que contiene el valor de la densidad de cada una de
las
%barras

%densidad(1,:)=2700;
%densidad(2,:)=2700;
%densidad(3,:)=2700;
%densidad(4,:)=2700;
%densidad(5,:)=2700;
%densidad(6,:)=2700;
%densidad(7,:)=2700;

%Aqui se introducen los nodos de cada de una de las barras

%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];

%En la variable SIZEBARRAS se guardar un vector que indicar la
dimensión de la matriz BARRAS
SIZEBARRAS=size(BARRAS);
%Aqui se crea una matriz con las coordenadas de los nodos, en la que
el número de fila se corresponder con el número del nodo. El primer
valor (primera columna)
%ser la coordenada 'x' y el segundo (segunda columna) la coordenada
'y'.

%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];

dimensionkestructura=size(Nodo);

%Matriz que nos dice las restricciones de cada nodo, tal que el primer
valor es desplazamiento horizontal, el segundo es desplazamiento
vertical
%y el tercero es giro. Y la primera fila corresponde al primer nodo,
la
%segunda al segundo y así sucesivamente. El valor cero indica que no
se
%produce desplazamiento del tipo que sea en ese nodo, mientras que el
uno
%indica que ese grado de libertad no está restringido y que por tanto
```

```
si se
%puede producir desplazamiento
%EL DOBLE DE NUMERO DE UNOS SON LAS CONDICIONES INICIALES QUE DEBEMOS
DE
%IMPONER AL ODE45 en el momento de la resolucion

%RestriccionesNodo(1,:)= [1 1];
%RestriccionesNodo(2,:)= [1 1];
%RestriccionesNodo(3,:)= [1 1];
%RestriccionesNodo(4,:)= [1 1];
%RestriccionesNodo(5,:)= [1 1];
%RestriccionesNodo(6,:)= [0 0];
%RestriccionesNodo(7,:)= [0 0];

contador=0;
for i=1;dimensionkestructura(1,1)
    for I=1:2
        if RestriccionesNodo(i,I)==0
            matdespl(i,I)=RestriccionesNodo(i,I);
        elseif RestriccionesNodo(i,I)==1
            contador=contador+1;
            matdespl(i,I)=RestriccionesNodo(i,I)*contador;
        end
    end
end

%Aqui se creara una matriz de una columna por tantas filas como barras
%tenemos, e indicaremos con el valor de la cte de rigidez del muelle
si es un barra-muelle y con un 0 si es
%una barra normal

%Barramuelle(1,:)= [0];
%Barramuelle(2,:)= [0];
%Barramuelle(3,:)= [1e6];
%Barramuelle(4,:)= [0];
%Barramuelle(5,:)= [0];
%Barramuelle(6,:)= [0];
%Barramuelle(7,:)= [0];
%Barramuelle(8,:)= [0];
%Barramuelle(9,:)= [1e6];

%Aqui se creara una matriz columna en la que diremos si la barra tiene
un
%amortiguador, dando el valor de cte de amortiguacion

%Barraamortiguador(1,:)= [0];
%Barraamortiguador(2,:)= [0];
%Barraamortiguador(3,:)= [0];
%Barraamortiguador(4,:)= [0];
%Barraamortiguador(5,:)= [0];
%Barraamortiguador(6,:)= [0];
%Barraamortiguador(7,:)= [0];
%Barraamortiguador(8,:)= [0];
%Barraamortiguador(9,:)= [0];

%Aqui introducimos la union del nodo menor con el mayor cambiando el
%segundo valor de NODOS, por el numero de nodo que une con el otro
for i=1:SIZEBARRAS(1,1)
    Barra(i,:)= [Nodo((BARRAS(i,1)),: ) Nodo((BARRAS(i,2)),: )];
end
```



```
barrassize=size(Barra);

%Bucle for e if que se calcula a partir de las coordenadas de los
nodos las
%longitudes de las barras, sus angulos y las k de cada barra
agrupandolas
%en una matriz de 6 filas X n columnas
for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
    angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=angulo(i,:);
        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end
%Para obtener las matrices de rigides en ejes globales de cada una de
las
%barras y muelles
for i=1:(barrassize(1,1))

    if (i==1) && (Barramuelle(i,1)==0)
        k=kglobalesarticuladasb(E(i,1),A(i,1),(L(1,i)),(alfa(i,1)));
    elseif (i==1) && (Barramuelle(i,1)~=0)
        k=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    elseif i>1 && (Barramuelle(i,1)==0)
        k(1:4, [(3*i)+(i-
3):4*i])=kglobalesarticuladasb(E(i,1),A(i,1),L(1,i),alfa(i,1));
    elseif (i>1) && (Barramuelle(i,1)~=0)
        k(1:4, [(3*i)+(i-
3):4*i])=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    end

end
%Para obtener las matrices de amortiguacion en ejes globales de cada
una de las barras
for i=1:(barrassize(1,1))
    if (i==1)

        camortig=kglobalesamortiguadorarticuladas(Barraamortiguador(i,1),alfa(
i,1));
    elseif (i>1)
        camortig(1:4, [(3*i)+(i-
3):4*i])=kglobalesamortiguadorarticuladas(Barraamortiguador(i,1),alfa(
```

```

i,1));
    end

end

dimensionkeestructura=size(Nodo);
dimensionK=dimensionkeestructura(1,1);

%Bucle para crear las matriz de ceros, unos y menos unos de las
matrices
%que conforman el sistema de ecuaciones
for i=1:dimensionK
    if i==1
        CEROS=zeros(2);
        UNOS=eye(2);
        UNOSN=-eye(2);
    elseif i>1
        %filas
        CEROS((i*2)-1:(i*2), [1:(2*i)])=zeros(2,(i*2));
        %columnas
        CEROS(1:(i+(i-2)), [(2*i)-1:(2*i)])=zeros((i+(i-2)),2);
        %matrices de unos y unos negativos
        UNOS((2*i)-1:(2*i), [(2*i)-1:(2*i)])=eye(2);
        UNOSN((2*i)-1:(2*i), [(2*i)-1:(2*i)])=-eye(2);
    end
end
%Bucle for e if que se calcula la matriz de rigidez en ejes globales

kglobales=CEROS;
for i=1:barrasize(1,1)
    if i==1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2, [1:2]);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2, [3:4]);
        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4, [1:2]);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-
1:2*n)+k(3:4, [3:4]);
    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2, [(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*m)-1:2*m,(2*m)-
1:2*m);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2, [(3*i)+(i-1):4*i])+kglobales((2*m)-1:2*m,(2*n)-1:2*n);
        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4, [(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*n)-1:2*n,(2*m)-
1:2*m);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-

```

```
1:2*n)+k(3:4, [(3*i)+(i-1):4*i])+kglobales((2*n)-1:2*n,(2*n)-1:2*n);  
  
end  
  
end  
  
K=kglobales;
```

%Bucle for e if que se calcula la matriz de rigidez en ejes globales

```
cglobales=CEROS;  
for i=1:barrassize(1,1)  
    if i==1  
        m=BARRAS(i,1);  
        n=BARRAS(i,2);  
        %m,m  
        cglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-  
1:2*m)+camortig(1:2, [1:2]);  
        %m,n  
        cglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-  
1:2*n)+camortig(1:2, [3:4]);  
        %n,m  
        cglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-  
1:2*m)+camortig(3:4, [1:2]);  
        %n,n  
        cglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-  
1:2*n)+camortig(3:4, [3:4]);  
  
        elseif i>1  
            m=BARRAS(i,1);  
            n=BARRAS(i,2);  
            %m,m  
            cglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-  
1:2*m)+camortig(1:2, [(3*i)+(i-3):(3*i)+(i-2)])+cglobales((2*m)-  
1:2*m,(2*m)-1:2*m);  
            %m,n  
            cglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-  
1:2*n)+camortig(1:2, [(3*i)+(i-1):4*i])+cglobales((2*m)-1:2*m,(2*n)-  
1:2*n);  
            %n,m  
            cglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-  
1:2*m)+camortig(3:4, [(3*i)+(i-3):(3*i)+(i-2)])+cglobales((2*n)-  
1:2*n,(2*m)-1:2*m);  
            %n,n  
            cglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-  
1:2*n)+camortig(3:4, [(3*i)+(i-1):4*i])+cglobales((2*n)-1:2*n,(2*n)-  
1:2*n);  
  
        end  
  
    end  
  
C=cglobales;
```

%Bucle for e if para ordenar en forma de columna las restricciones de los  
%nodos de forma que en despvel quedan recogidas las gdl de cada  
%nodo y de donde podremos sacar la reducida a la que llamaremos  
%despvelred

```
for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif (i==1)
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==1)
                desplazamiento((i*2)-1,1)=((i*2)-
1)*(RestriccionesNodo(i,I));
                elseif (i>1) && (I==2)

desplazamiento((i*2),1)=(i*2)*(RestriccionesNodo(i,I));

        end
    end
end

sizedesplazamiento=size(desplazamiento);

for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)

velocidades=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
        elseif i==1

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i
,I);
            elseif (i>1) && (I==1)
                velocidades((i*2)-1,1)=(((i*2)-
1)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
                elseif (i>1) && (I==2)

velocidades((i*2),1)=((i*2)+sizedesplazamiento(1,1))*(RestriccionesNod
o(i,I));

        end
    end
end

%despvel aporta el vector columna total con los
%desplazamientos y las velocidades

despvel=[desplazamiento;velocidades];
sizedespvel=size(despvel);

%Bucle para obtener la posicion de la primera incognita de
desplazamientos
%y velocidades en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred===-1)
        despvelred=[despvel(i,1)];
    end
end

end
```

%Con esto se obtiene un vector columna con las posiciones de las velocidades y desplazamientos que son incognita

```
dvr=despvelred;
```

```
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end
```

```
sizedespvelred=size(despvelred);
```

%Obtencion de la matriz de masas

```
MASAS=zeros(2,dimensionkestructura(1,1)*2);
```

```
for i=1:(barrassize(1,1))
    mas=densidad(i,1)*A(i,1)*L(1,i);
    for I=1:2

        MASAS(1:2, [(2*BARRAS(i,I))-1:2*BARRAS(i,I)])=(MASAS(1:2,
[(2*BARRAS(i,I))-1:2*BARRAS(i,I)])+[mas/2 0 ;0 (mas/2)]);

    end
end
```

```
for i=1:dimensionK
    if i==1
        M=MASAS(1:2, [1:2]);
    elseif i>1

        M(((i*2)-1):(i*2), [((i*2)-1):(i*2)])=MASAS(1:2, [((i*2)-
1):(i*2)]);

    end
end
```

%Matrices que plantean el sistema de ecuaciones

```
h=[CEROS UNOSN;K C];
g=[UNOS CEROS;CEROS M];
```

%Para la obtencion de las matrices reducidas que contienen a la matriz de rigidez y a la matriz de masas se emplea el siguiente bucle

```
for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[h(despvelred(i,1),despvelred(I,1))];
            G=[g(despvelred(i,1),despvelred(I,1))];
        elseif i>0
```

```
H(i,I)=h(despvelred(i,1),despvelred(I,1));
G(i,I)=g(despvelred(i,1),despvelred(I,1));
    end
end
end

%Matriz inversa de G

O=inv(G);
```

## Anexo 3

En este anexo se encuentran los ficheros con los que es posible obtener los valores y modos propios de estructuras articuladas y reticuladas.

### Cálculo valores y modos propios de una estructura articulada

```
function
ValoresPropiosEstructuraArticulada=valores_propios_estructura_articulada(E,A,densidad,BARRAS,Nodo,RestriccionesNodo,Barramuelle)

%E=70e9;
%A=pi*((1e-2)/2)^(2);

%densidad=2700;

%aquí introducimos los nodos de cada una de las barras;

%BARRAS=[1 2;1 3;1 7;2 3;2 4;3 4;3 5;4 5;5 6];
SIZEBARRAS=size(BARRAS);

%aquí introducimos las coordenadas de los nodos, nos fijamos en la primera
%columna de
%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];
%Nodo(6,:)= [1 -0.5];
%Nodo(7,:)= [0 -0.5];

dimensionestructura=size(Nodo);

%matriz que nos dice las restricciones de cada nodo, tal que el primer
%valor es desplazamiento horizontal, el segundo es desplazamiento
vertical
%y el tercero es giro. Y la primera fila corresponde al primer nodo,
la
%segunda al segundo y así sucesivamente. El valor cero indica que no
se
%produce desplazamiento del tipo que sea en ese nodo, mientras que el
uno
%indica que ese grado de libertad no está restringido y que por tanto
sí se
%puede producir desplazamiento
```

```
%EL DOBLE DE NUMERO DE UNOS SON LAS CONDICIONES INICIALES QUE DEBEMOS
DE
%IMPONER AL ODE45 en el momento de la resolucio
%RestriccionesNodo(1,:)= [1 1];
%RestriccionesNodo(2,:)= [1 1];
%RestriccionesNodo(3,:)= [1 1];
%RestriccionesNodo(4,:)= [1 1];
%RestriccionesNodo(5,:)= [1 1];
%RestriccionesNodo(6,:)= [0 0];
%RestriccionesNodo(7,:)= [0 0];

%introducimos en estos vectores el valor de las fuerzas aplicadas y en
que
%sentido en cada uno de los nodos      SUSTITUIR EL 1 POR FUERZA

%FUERZA*((2)^(0.5))/2
%FuerzasNodo(1,:)= [0 0];
%FuerzasNodo(2,:)= [0 0];
%FuerzasNodo(3,:)= [0 -1000];
%FuerzasNodo(4,:)= [0 0];
%FuerzasNodo(5,:)= [0 0];
%FuerzasNodo(6,:)= [0 0];
%FuerzasNodo(7,:)= [0 0];

%aquí crearemos una matriz de una columna por tantas filas como barras
%tenemos, e indicaremos con el valor de la cte de rigidez del muelle
si es un barra-muelle y con un 0 si es
%una barra normal
%Barramuelle(1,:)= [0];
%Barramuelle(2,:)= [0];
%Barramuelle(3,:)= [1e6];
%Barramuelle(4,:)= [0];
%Barramuelle(5,:)= [0];
%Barramuelle(6,:)= [0];
%Barramuelle(7,:)= [0];
%Barramuelle(8,:)= [0];
%Barramuelle(9,:)= [1e6];

%aquí introducimos la unión del nodo menor con el mayor cambiando el
%segundo valor de NODOS, por el número de nodo que une con el otro
for i=1:SIZEBARRAS(1,1)
    Barra(i,:)= [Nodo((BARRAS(i,1)),:), Nodo((BARRAS(i,2)),:)]';
end

barrassize=size(Barra);

%bucle for e if que se calcula a partir de las coordenadas de los
nodos las
%longitudes de las barras, sus ángulos y las k de cada barra
agrupandolas
%en una matriz de 6 filas X n columnas

for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
```

```

L(1,i)=(((a(i,1))^(2))+((a(i,2))^(2)))^(0.5);
angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end

%para obtener las matrices de rigidez de cada una de las barras
for i=1:(barrassize(1,1))
    if (i==1) && (Barramuelle(i,1)==0)
        k=kglobalesarticuladasb(E(i,1),A(i,1),(L(1,i)),(alfa(i,1)));
    elseif (i==1) && (Barramuelle(i,1)~=0)
        k=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    elseif i>1 && (Barramuelle(i,1)==0)
        k(1:4, [(3*i)+(i-3):4*i])=kglobalesarticuladasb(E(i,1),A(i,1),L(1,i),alfa(i,1));
    elseif (i>1) && (Barramuelle(i,1)~=0)
        k(1:4, [(3*i)+(i-3):4*i])=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    end
end

dimensionkestructura=size(Nodo);
dimensionK=dimensionkestructura(1,1);
for i=1:dimensionK
    if i==1
        CEROS=zeros(2);
        UNOS=eye(2);
        UNOSN=-eye(2);
    elseif i>1
        %filas
        CEROS(((i*2)-1):(i*2), [1:(2*i)])=zeros(2,(i*2));
        %columnas
        CEROS(1:(i+(i-2)), [((2*i)-1):(2*i)])=zeros((i+(i-2)),2);
        %matrices de unos y unos negativos
        UNOS(((2*i)-1):(2*i), [((2*i)-1):(2*i)])=eye(2);
        UNOSN(((2*i)-1):(2*i), [((2*i)-1):(2*i)])=-eye(2);
    end
end

```



```
%bucle for e if que se calcula la matriz de rigidez en ejes globales
kglobales=CEROS;
for i=1:barrasize(1,1)
    if i==1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2,[1:2]);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2,[3:4]);
        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4,[1:2]);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-
1:2*n)+k(3:4,[3:4]);

    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2,[(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*m)-1:2*m,(2*m)-
1:2*m);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2,[(3*i)+(i-1):4*i])+kglobales((2*m)-1:2*m,(2*n)-1:2*n);
        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4,[(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*n)-1:2*n,(2*m)-
1:2*m);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-
1:2*n)+k(3:4,[(3*i)+(i-1):4*i])+kglobales((2*n)-1:2*n,(2*n)-1:2*n);

end

end

K=kglobales;
```

```
%bucle for e if para ordenarme en forma de columna las restricciones
de los
%nodos de forma que en Restricciondesp quedan recogidas las gdl de
cada
%nodo y de donde podremos sacar la reducida a la que llamaremos
%Restriccionreducida
```

```
for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif (i==1)
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==1)
            desplazamiento((i*2)-1,1)=((i*2)-
1)*(RestriccionesNodo(i,I));
```

```
elseif (i>1) && (I==2)

desplazamiento((i*2),1)=(i*2)*(RestriccionesNodo(i,I));

    end
end

sizedesplazamiento=size(desplazamiento);

for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)

velocidades=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
            elseif i==1

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
            elseif (i>1) && (I==1)
                velocidades((i*2)-1,1)=(((i*2)-1)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
            elseif (i>1) && (I==2)

velocidades((i*2),1)=((i*2)+sizedesplazamiento(1,1))*(RestriccionesNodo(i,I));

        end
    end
end

%Restricciondespvel nos aporta el vector columna total con los
%desplazamientos y las velocidades

despvel=[desplazamiento;velocidades];
sizedespvel=size(despvel);

%bucle para obtener la posicion de la primera incognita de
desplazamientos
%y velocidades en la reducida

despvelred=-1;
for i=1:(sizedespvel(1,1))/2
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end

%con esto obtenemos un vector columna con las posiciones de las
velocidades y desplazamientos que son incognita

dvr=despvelred;

for i=1:(sizedespvel(1,1))/2
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end
```

```
end
end

sizedespvelred=size(despvelred);

%para poder determinar el numero de nodos compartidos y asi conocer la
%distribucion de masas en las barras

for i=1:(barrassize(1,1))
    if i==1
        BARRASH=BARRAS(i,:);
    elseif i>1
        BARRASH(1, [i+(i-1):2*i])=BARRAS(i,:);
    end
end

%esto nos aportara un vector, el vector nodoscompartidos que nos dira
%por cuanto hemos de multiplicar la matriz de masa de cada nodo

masaspornodo=size(BARRASH);
nodoscompartidos=zeros(1,dimensionkestructura(1,1));
for i=1:dimensionkestructura(1,1)
    for I=1:masaspornodo(1,2)
        if i==BARRASH(1,I)
            nodoscompartidos(1,i)=(nodoscompartidos(1,i))+1;
        end
    end
end

%matriz de masas de 3 filas por n columnas que variara en funcion del
%numero de nodos

for i=1:(barrassize(1,1))
    mas=densidad(i,1)*A(i,1)*L(1,i);
    for I=1:2
        %[(m/2) 0 0;0 (m/2) 0;0 0 ((m*((L(1,i))^(2)))/3)]
        if (i==1) && (I==1)
            MASAS=[(mas/2) 0;0 (mas/2)];
        elseif i>=1
            MASAS(1:2, [(2*BARRAS(i,I))-1:2*BARRAS(i,I)])=[(mas/2) 0 ;0
(mas/2)];
        end
    end
end

for i=1:dimensionK
    if i==1
        M=MASAS(1:2, [1:2])*nodoscompartidos(1,i);
    elseif i>1
        M(((i*2)-1):(i*2), [((i*2)-1):(i*2)])=MASAS(1:2, [((i*2)-
1):(i*2)])*nodoscompartidos(1,i);
    end
end
```

%para la obtencionde las matrices reducidas que contienen a la matriz de  
%rigidez y a la matriz de masas se emplea el siguiente bucle

```
for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[K(despvelred(i,1),despvelred(I,1))];
            G=[M(despvelred(i,1),despvelred(I,1))];
        elseif i>0
            H(i,I)=K(despvelred(i,1),despvelred(I,1));
            G(i,I)=M(despvelred(i,1),despvelred(I,1));
        end
    end
end
```

```
N=inv(G);
```

```
MK=H*N;
```

%el autovector asociado al autovalor D(j,j) es P(:,j)

```
[P,D]=eig(MK);
```

```
sizeD=size(D);
```

```
for i=1:sizeD(1,1)
    Autovalorescolumna(i,:)=real(D(i,i)^(0.5));
end
```

```
Autovectoresenfilas=P';
```

```
ValoresPropiosEstructuraArticulada=[Autovalorescolumna
Autovectoresenfilas];
```

### Cálculo valores y modos propios de una estructura reticulada

```
function
ValoresPropiosEstructuraReticulada=valores_propios_estructura_reticul
ada(E,A,I,densidad,Nodo,BARRAS,RestriccionesNodo,Barramuelle)
```

```
%E=70e9;
%A=pi*(((1e-2)/2)^(2));
%I=pi*(((1e-2)/2)^(4))/2;
```

```
%densidad=2700;
%densidad(1,:)=2700;
%densidad(2,:)=2700;
%densidad(3,:)=2700;
%densidad(4,:)=2700;
%densidad(5,:)=2700;
%densidad(6,:)=2700;
%densidad(7,:)=2700;
%densidad(8,:)=2700;
%densidad(9,:)=2700;
```

%aqui introducimos los nodos de cada de una de las barras

```
%BARRAS=[1 2;1 3;1 7;2 3;2 4;3 4;3 5;4 5;5 6];
SIZEBARRAS=size(BARRAS);

%aquí introducimos las coordenadas de los nodos, pero nos tenemos que
fijar
%en la primera columna de la matriz BARRAS, e ir poniendo como primera
%coordenada Nodo((BARRAS(primer coordenada,1)), :) el número de fila
en que
%se encuentra ese número, sin repetirlo dos veces
%Nodo(1,:)= [0 0];
%Nodo(2,:)= [-0.353 -0.853];
%Nodo(3,:)= [0 -0.5];
%Nodo(4,:)= [0.353 -0.853];
%Nodo(5,:)= [1 0];
%Nodo(6,:)= [1 -0.5];
%Nodo(7,:)= [0 -0.5];

%aquí introducimos la unión del nodo menor con el mayor formando una
matriz
%de tantas filas como barras tenga la estructura y cuyos dos primeros
%valores son las coordenadas del nodo menor, y las otras dos valores
son
%las coordenadas del nodo mayor
for i=1:SIZEBARRAS(1,1)
    Barra(i,:)= [Nodo((BARRAS(i,1)), :) Nodo((BARRAS(i,2)), :)];
end
barrasize=size (Barra);
%aquí crearemos una matriz de una columna por tantas filas como barras
%tenemos, e indicaremos con el valor de la cte de rigidez del muelle
si es una barra-muelle y con un 0 si es
%una barra normal
%Barramuelle(1,:)= [0];
%Barramuelle(2,:)= [0];
%Barramuelle(3,:)= [1e6];
%Barramuelle(4,:)= [0];
%Barramuelle(5,:)= [0];
%Barramuelle(6,:)= [0];
%Barramuelle(7,:)= [0];
%Barramuelle(8,:)= [0];
%Barramuelle(9,:)= [1e6];

%matriz que nos dice las restricciones de cada nodo, tal que el primer
%valor es desplazamiento horizontal, el segundo es desplazamiento
vertical
%y el tercero es giro. Y la primera fila corresponde al primer nodo,
la
%segunda al segundo y así sucesivamente. El valor cero indica que no
se
%produce desplazamiento del tipo que sea en ese nodo, mientras que el
uno
%indica que ese grado de libertad no está restringido y que por tanto
si se
%puede producir desplazamiento
%CONTAR CUANTOS UNOS HAY PORQUE LE IMPONDREMOS TANTAS CONDICIONES DE
%CONTORNO EN EL ODE45 COMO NUMEROS DE UNOS MULTIPLICADOS POR DOS
%TENGAMOS.

%RestriccionesNodo(1,:)= [1 1 1];
```

```
%RestriccionesNodo(2,:)= [1 1 1];
%RestriccionesNodo(3,:)= [1 1 1];
%RestriccionesNodo(4,:)= [1 1 1];
%RestriccionesNodo(5,:)= [1 1 1];
%RestriccionesNodo(6,:)= [0 0 1];
%RestriccionesNodo(7,:)= [0 0 1];

dimensionkeestructura=size(Nodo);

%bucle for e if que se calcula a partir de las coordenadas de los
nodos las
%longitudes de las barras, sus angulos y las k de cada barra
agrupandolas
%en una matriz de 6 filas X n columns

for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
    angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end

for i=1:(barrassize(1,1))
    if (i==1) && (Barramuelle(i,1)==0)

k=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alfa(i,1)));
        elseif (i==1) && (Barramuelle(i,1)~=0)
            k=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        elseif i>1 && (Barramuelle(i,1)==0)
            k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alf
a(i,1)));
        elseif (i>1) && (Barramuelle(i,1)~=0)
            k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        end
end
```

```
dimensionK=dimensionkestructura(1,1);
for i=1:dimensionK
    if i==1
        CEROS=zeros(3);
        UNOS=eye(3);
        UNOSN=-eye(3);
    elseif i>1
        %filas
        CEROS(((i*3)-2):(i*3), [1:(3*i)])=zeros(3,(i*3));
        %columnas
        CEROS(1:(i+(i-3)), [((3*i)-2):(3*i)])=zeros((i+(i-3)),3);
        %matrices de unos y unos negativos
        UNOS((((2*i)+(i-2)):(3*i)), [((2*i)+(i-2)):(3*i)])=eye(3);
        UNOSN((((2*i)+(i-2)):(3*i)), [((2*i)+(i-2)):(3*i)])=-
eye(3);
    end
end
```

```
%bucle for e if que se calcula la matriz de rigidez en ejes globales
kglobales=CEROS;
```

```
for i=1:(barrasize(1,1))
    if (i==1)
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m])+k(1:3, [1:3]);
        %m,n
        kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n])+k(1:3, [4:6]);
        %n,m
        kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m])+k(4:6, [1:3]);
        %n,n
        kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n, [(2*n)+(n-2):3*n])+k(4:6, [4:6]);
```

```
    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m]))+(k(1:3, [(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m]));
        %m,n
        kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n]))+(k(1:3, [(5*i)+(i-
2):6*i]))+(kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n]));
        %n,m
        kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m]))+(k(4:6, [(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m]));
```

```

        %n,n
        kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n,[(2*n)+(n-2):3*n]))+(k(4:6, [(5*i)+(i-
2):6*i]))+(kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n]));

end

end

K=kglobales;

%bucle for e if para ordenarme en forma de columna las restricciones
de los
%nodos de forma que en Restricciondesp quedan recogidas las gdl de
cada
%nodo y de donde podremos sacar la reducida a la que llamaremos
%Restriccionreducida

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif i==1
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==1)
            desplazamiento((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==2)
            desplazamiento((i*I)+(i-1),1)=((i*I)+(i-
1))*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==3)

desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
            end
        end
    end

    sizedesplazamiento=size(desplazamiento);

    for i=1:max(max(BARRAS))
        for I=1:3
            if (i==1) && (I==1)

velocidades=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i,I);
                elseif i==1

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*RestriccionesNodo(i
,I);
                    elseif (i>1) && (I==1)
                        velocidades((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))+sizedesplazamiento(1,1))*(RestriccionesNodo(i,I));
                    elseif (i>1) && (I==2)
                        velocidades((i*I)+(i-1),1)=((i*I)+(i-
1))+sizedesplazamiento(1,1))*(RestriccionesNodo(i,I));
                    elseif (i>1) && (I==3)

velocidades(i*I,1)=((i*I)+sizedesplazamiento(1,1))*(RestriccionesNodo(
i,I));
                        end
                    end
                end
            end
        end
    end
end

```



```
%Restricciondespvel nos aporta el vector columna total con los
%desplazamientos y las velocidades

despvel=[desplazamiento;velocidades];
sizedespvel=size(despvel);

%bucle para obtener la posicion de la primera incognita de
desplazamientos
%y velocidades en la reducida

despvelred=-1;
for i=1:(sizedespvel(1,1))/2
    if (despvel(i,1)~=0) && (despvelred== -1)
        despvelred=[despvel(i,1)];
    end
end

%con esto obtenemos un vector columna con las posiciones de las
velocidades y desplazamientos que son incognita

dvr=despvelred;

for i=1:(sizedespvel(1,1))/2
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end

sizedespvelred=size(despvelred);

%para poder determinar el numero de nodos compartidos y asi conocer la
%distribucion de masas en las barras

for i=1:(barrassize(1,1))
    if i==1
        BARRASH=BARRAS(i,:);
    elseif i>1
        BARRASH(1, [i+(i-1):2*i])=BARRAS(i,:);
    end
end

%esto nos aportara un vector, el vector nodoscompartidos que nos dira
%por cuanto hemos de multiplicar la matriz de masa de cada nodo

masaspornodo=size(BARRASH);
nodoscompartidos=zeros(1,dimensionkestructura(1,1));
for i=1:dimensionkestructura(1,1)
    for I=1:masaspornodo(1,2)
        if i==BARRASH(1,I)
            nodoscompartidos(1,i)=(nodoscompartidos(1,i))+1;
        end
    end
end
```

```

        end
    end
end

%matriz de masas de 3 filas por n columnas que variara en funcion del
%numero de nodos

for i=1:(barrassize(1,1))
    mas=densidad(i,1)*A(i,1)*L(1,i);
    for I=1:2
        %[(m/2) 0 0;0 (m/2) 0;0 0 ((m*((L(1,i))^(2)))/3)]
        if (i==1) && (I==1)
            MASAS=[(mas/2) 0 0;0 (mas/2) 0;0 0 ((mas*((L(1,i))^(2)))/3)];
        elseif i>=1
            MASAS(1:3, [(2*BARRAS(i,I))+(BARRAS(i,I)-
2):3*BARRAS(i,I)])=[(mas/2) 0 0;0 (mas/2) 0;0 0
((mas*((L(1,i))^(2)))/3)];
        end
    end
end

end

for i=1:dimensionK
    if i==1
        M=MASAS(1:3, [1:3])*nodoscompartidos(1,i);
    elseif i>1
        M(((i*3)-2):(i*3), [((i*3)-2):(i*3)])=MASAS(1:3, [((i*3)-
2):(i*3)])*nodoscompartidos(1,i);

    end
end

end

%para la obtencionde las matrices reducidas que contienen a la matriz
de
%rigidez y a la matriz de masas se emplea el siguiente bucle

for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[K(despvelred(i,1),despvelred(I,1))];
            G=[M(despvelred(i,1),despvelred(I,1))];
        elseif i>0
            H(i,I)=K(despvelred(i,1),despvelred(I,1));
            G(i,I)=M(despvelred(i,1),despvelred(I,1));
        end
    end
end

end

N=inv(G);

MK=H*N;

%el autovector asociado al autovalor D(j,j) es P(:,j)

[P,D]=eig(MK);

```

```
%con los siguientes bucles vamos a generar una columna que nos
indicara la
%posicion del autovector en la matriz D(de autovectores) y que sera
%finalmente lo que saldra en el interface

sizeD=size(D);

for i=1:sizeD(1,1)
    Autovalorescolumna(i,:)=real(D(i,i)^(0.5));
end

Autovectoresenfilas=P';

%contadorrestricc=0;
%for i=1:dimensionkestructura(1,1)
%    for I=1:2
%        if (RestriccionesNodo(i,I)==1)
%            contadorrestricc=contadorrestricc+1;
%            ColumnaModospropios(contadorrestricc,:)=contadorrestricc;
%            elseif (RestriccionesNodo(i,I)~=0)
%                ColumnaModospropios;
%            end
%        end
%    end
%end

ValoresPropiosEstructuraReticulada=[Autovalorescolumna
Autovectoresenfilas];
```

## Anexo 4

En este anexo se encuentran los ficheros que permiten obtener los desplazamientos de en estructuras articuladas y reticuladas sometidas a cargas estáticas y dinámicas.

### Obtención de desplazamientos para estructuras articuladas en estático.

```
function
DesplazamientoXarticulada=Desplzamientosenestaticoestructuraarticulada
(E,A,BARRAS,Nodo,RestriccionesNodo,FuerzaNodos,Barramuelle)

%TODOS LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
%INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA

%Vector que contiene el valor del mÓdulo de elasticidad de cada barra

%E(1,:)=70e9;
%E(2,:)=70e9;
%E(3,:)=70e9;
%E(4,:)=70e9;
%E(5,:)=70e9;
%E(6,:)=70e9;
%E(7,:)=70e9;

%Vector que contiene el valor del Área de cada barra

%A(1,:)=pi*(((1e-2)/2)^(2));
```

```
%A(2,:)=pi*(((1e-2)/2)^(2));
%A(3,:)=pi*(((1e-2)/2)^(2));
%A(4,:)=pi*(((1e-2)/2)^(2));
%A(5,:)=pi*(((1e-2)/2)^(2));
%A(6,:)=pi*(((1e-2)/2)^(2));
%A(7,:)=pi*(((1e-2)/2)^(2));

%Aqui se introducen los nodos de cada de una de las barras

%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];

%En la variable SIZEBARRAS se guardar· un vector que indicar· la
dimensi·n de la matriz BARRAS

SIZEBARRAS=size(BARRAS);

%Aqui se crea una matriz con las coordenadas de los nodos, en la que
el
n·mero de fila se corresponder· con el n·mero del nodo. El primer
valor (primera columna)
ser· la coordenada 'x' y el segundo (segunda columna) la coordenada
'y'.

%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];

%La dimensi·n de la matriz Nodo, se memorizara en la variable
'dimensionkeestructura' para c·lculos posteriores.

dimensionkeestructura=size(Nodo);
dimensionK=dimensionkeestructura(1,1);

%La matriz RestriccionesNodo tendr· dos columnas y habr· que
introducir
%las restricci·nes de cada nodo por filas, siendo el primer valor, la
restricci·n del
%desplazamiento horizontal y el tercero el giro.
%Si se trata de un grado restringido se indicara con un '0' y si por
el
%contrario es un grado de libertad no restringido, se indicar· con un
'1'.
%Las condiciones de contorno vendr·n determinadas por las
restricciones de
%los nodos, que habr· tantas como n·mero de unos por dos tenga la
matriz.

%RestriccionesNodo(1,:)= [0 0]; %restriccion del nodo 1 =
[desplazamiento horizontal nulo    desplazamiento vertical nulo]
%RestriccionesNodo(2,:)= [1 1];
%RestriccionesNodo(3,:)= [1 1];
%RestriccionesNodo(4,:)= [1 1];
%RestriccionesNodo(5,:)= [0 0];

%La Matriz FuerzaNodos recoge la amplitud de la fuerza senoidal que se
va a
%aplicar, tiene 2 columnas, cada una se corresponde con los valore [x
y],
%y el n·mero de fila se corresponde con el n·mero de nodo.
```

```
%Fuerzanodos(1,:)=[0 0];
%Fuerzanodos(2,:)=[0 0];
%Fuerzanodos(3,:)=[0 -1000];
%Fuerzanodos(4,:)=[0 0];
%Fuerzanodos(5,:)=[0 0];

%Bucle para obtener una matriz como la anterior, pero con los valores
de
%las fuerzas senoidales aplicadas a cada nodo.

for i=1:dimensionkestructura(1,1)
FuerzasNodo(i,:)=FuerzaNodos(i,:);
end

%Aqui se crear un vector dispuesto en forma de columna, con tantas
filas
%como barras tenga la estructura y en la que cada número de fila se
%corresponder con el número de la barra. Asignándose el valor de la
%constantes de rigidez en caso de que esa barra se considere un
muelle.

%Barramuelle(1,:)=0;
%Barramuelle(2,:)=0;
%Barramuelle(3,:)=0;
%Barramuelle(4,:)=0;
%Barramuelle(5,:)=0;
%Barramuelle(6,:)=0;
%Barramuelle(7,:)=0;

%Mediante el siguiente bucle, se generar la matriz 'Barra', en el que
cada
número de fila se corresponder con el número de barra. Cada fila
tendr 4 valores, los dos primeros serán
%las coordenadas del nodo menor de la barra y las otras dos las
coordenadas
%del nodo mayor.

for i=1:SIZEBARRAS(1,1)
Barra(i,:)=[Nodo((BARRAS(i,1)),:)-Nodo((BARRAS(i,2)),:)];
end

%La dimensión de la matriz 'Barra', se memorizar en la variable
%'barrassize' para cálculos posteriores.

barrassize=size(Barra);

%Bucle for e if con el que se calcula a partir de las coordenadas de
los nodos las
%longitudes de las barras, sus angulos

for i=1:(barrassize(1,1))
a(i,1)=(Barra(i,3))-(Barra(i,1));
a(i,2)=(Barra(i,4))-(Barra(i,2));
L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
angulo(i,:)=atan((a(i,2))/(a(i,1)));
if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
alfa(i,:)=0;
elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

alfa(i,:)=angulo(i,:);
elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
```

```

        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end

%Bucle a partir del cual se calculan las matrices de rigidez de cada
barra en ejes globales agrupandolas
%en una matriz de 6 filas X n columnas. Posteriormente se utilizará la
%matriz 'k' para el cálculo de la matriz global de la estructura.

for i=1:(barrassize(1,1))
    if (i==1) && (Barramuelle(i,1)==0)
        k=kglobalesarticuladasb(E(i,1),A(i,1),(L(1,i)),(alfa(i,1)));
    elseif (i==1) && (Barramuelle(i,1)~=0)
        k=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    elseif i>1 && (Barramuelle(i,1)==0)
        k(1:4, [(3*i)+(i-
3):4*i])=kglobalesarticuladasb(E(i,1),A(i,1),L(1,i),alfa(i,1));
    elseif (i>1) && (Barramuelle(i,1)~=0)
        k(1:4, [(3*i)+(i-
3):4*i])=kglobalesmuellearticuladas(Barramuelle(i,1),alfa(i,1));
    end
end

end

%Obtención de las matriz de ceros

for i=1:dimensionK
    if i==1
        CEROS=zeros(2);
    elseif i>1
        %filas
        CEROS(((i*2)-1):(i*2), [1:(2*i)])=zeros(2,(i*2));
    end
end

%Bucle con el que se calcula la matriz de rigidez en ejes globales de
%nombre 'kglobales'

kglobales=CEROS;
for i=1:barrassize(1,1)
    if i==1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2, [1:2]);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2, [3:4]);
    end
end

```

```

        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4,[1:2]);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-
1:2*n)+k(3:4,[3:4]);

    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)-1:2*m,(2*m)-1:2*m)=CEROS((2*m)-1:2*m,(2*m)-
1:2*m)+k(1:2,[(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*m)-1:2*m,(2*m)-
1:2*m);
        %m,n
        kglobales((2*m)-1:2*m,(2*n)-1:2*n)=CEROS((2*m)-1:2*m,(2*n)-
1:2*n)+k(1:2,[(3*i)+(i-1):4*i])+kglobales((2*m)-1:2*m,(2*n)-1:2*n);
        %n,m
        kglobales((2*n)-1:2*n,(2*m)-1:2*m)=CEROS((2*n)-1:2*n,(2*m)-
1:2*m)+k(3:4,[(3*i)+(i-3):(3*i)+(i-2)])+kglobales((2*n)-1:2*n,(2*m)-
1:2*m);
        %n,n
        kglobales((2*n)-1:2*n,(2*n)-1:2*n)=CEROS((2*n)-1:2*n,(2*n)-
1:2*n)+k(3:4,[(3*i)+(i-1):4*i])+kglobales((2*n)-1:2*n,(2*n)-1:2*n);

end

end

%Se le asigna la letra 'K' los valores que guarda la palabra
'kglobales'

K=kglobales;

%Estos dos bucles sirven para ordenar en forma de columna las
restricciones de los
%nodos de forma de forma que indique la posici n del vector en el que
el
%grado no esta restringido y mediante lo que se podr  obtener las
matrices
%reducidas para plantear el sistema de ecuaciones a resolver.

for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif (i==1)
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==1)
            desplazamiento((i*2)-1,1)=((i*2)-
1)*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==2)

desplazamiento((i*2),1)=(i*2)*(RestriccionesNodo(i,I));

        end
    end
end

sizedesplazamiento=size(desplazamiento);

%Se crea un ´nico vector (el que iria multiplicado por la matriz que

```

```
contiene a
%la matriz de rigidez y de amortiguamiento) en el que la primera mitad
de valores se
%corresponden con los desplazamientos y la segunda con las velocidades

despvel=[desplazamiento];
sizedespvel=size(despvel);

%Bucle para obtener la posicion de la primera incognita de
desplazamientos
%y velocidades en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end

%Con esto se obtiene un vector columna con las posiciones de las
%velocidades y desplazamientos que son incognita

dvr=despvelred;

for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end

sizedespvelred=size(despvelred);

%Para la obtencion de las matrices reducidas que contienen a la matriz
de
%rigidez y a la matriz de masas se emplea el siguiente bucle

for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[K(despvelred(i,1),despvelred(I,1))];
        elseif i>0
            H(i,I)=K(despvelred(i,1),despvelred(I,1));
        end
    end
end

%Obtencion de la columna de las fuerzas aplicadas en los nodos

for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)
            fuerzaaar=FuerzasNodo(i,I);
        elseif i>-1
            fuerzaaar=[fuerzaaar;FuerzasNodo(i,I)];
        end
    end
end
```



```
sizefuerza=size(fuerzaaar);  
  
FUERZASCOLUMNA=[fuerzaaar];  
  
for i=1:sizedespvelred(1,1)  
    if i==1  
        fuerzared=FUERZASCOLUMNA(despvelred(i,1),1);  
    elseif i>1  
        fuerzared=[fuerzared;FUERZASCOLUMNA(despvelred(i,1),1)];  
    end  
end  
  
%Obtenci n de desplazamientos de la estructura.  
  
DesplazamientoXarticulada=H\fuerzared;
```

### Obtenci n de desplazamientos para estructuras articuladas en din mico.

```
function  
dx=xprimaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ(t,x,O,H,t  
fin,FuerzaNodos,BARRAS,despvelred,omegaaa,desfase,dimensionkestructura  
ramuelle,Barraamortiguador,tfin,desfase,alfa,L)  
  
%Vector desfase con tantas filas como nodos tenga la estructura, sera  
ua de  
%las componentes de la fuerza y habra que indicar en el caso de que en  
u  
%determinado nodo se aplique una fuerza senoidal, el valor de la  
frecuencia  
  
%desfase(1,:)=0;  
%desfase(2,:)=0;  
%desfase(3,:)=1000;  
%desfase(4,:)=0;  
%desfase(5,:)=0;  
  
%Vector omega con tantas filas como nodos tenga la estructura, sera ua  
de  
%las componentes de la fuerza y habra que indicar en el caso de que en  
u  
%determinado nodo se aplique una fuerza senoidal, el valor de la  
frecuencia  
  
%omegaaa(1,:)=0;  
%omegaaa(2,:)=0;  
%omegaaa(3,:)=1000;  
%omegaaa(4,:)=0;  
%omegaaa(5,:)=0;  
  
%Puesto que la fuerza depende del tiempo, todos las formulas que  
contengan  
%el tiempo, deben de encontrarse en el fichero que va a ser resultado por  
el  
%ode45.  
  
sizedespvelred=size(despvelred);  
  
%Aqui se crea el vector 'FUERZA' en el que cada uno de los elementos  
del  
%vector ser  el resultado de esta operacion
```

```
"sin((omegaaa(i,:)*t)+desfase(i,:))"

for i=1:dimensionkestructura(1,1)
FUERZA(i,:)=sin((omegaaa(i,:)*t)+desfase(i,:));
end

%Bucle para obtener una matriz como la anterior, pero con los valores
de
%las fuerzas senoidales aplicadas a cada nodo.

for i=1:dimensionkestructura(1,1)
    for I=1:2
        FuerzasNodo(i,I)=FuerzaNodos(i,I)*FUERZA(i,1);
    end
end

%Obtencion de la columna de las fuerzas aplicadas en los nodos

for i=1:max(max(BARRAS))
    for I=1:2
        if (i==1) && (I==1)
            fuerzaaar=FuerzasNodo(i,I);
        elseif i>-1
            fuerzaaar=[fuerzaaar;FuerzasNodo(i,I)];
        end
    end
end

sizefuerza=size(fuerzaaar);

cerosfuerza=zeros(sizefuerza(1,1),1);

%Aqui y se obtiene la culumna de fuerzas aplicadas a la estructura

FUERZASCOLUMNA=[cerosfuerza;fuerzaaar];

%Aqui se obtiene la columna reducida de las fuerzas aplicadas a la
%estructura

for i=1:sizedespvelred(1,1)
    if i==1
        fuerzared=FUERZASCOLUMNA(despvelred(i,1),1);
    elseif i>1
        fuerzared=[fuerzared;FUERZASCOLUMNA(despvelred(i,1),1)];
    end
end

(t*100)/tfin

dx=(O*fuerzared)-(O*(H*x));
```

### Obtención de desplazamientos para estructuras reticuladas en estático.

```
function
DesplazamientoXreticulada=Desplazamientosenestaticoestructurareticulad
a(E,A,I,BARRAS,Nodo,RestriccionesNodo,FuerzaNodos,Barramuelle)

%TODO LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
%INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA
```

%Vector que contiene el valor del módulo de elasticidad de cada barra

```
%E(1,:)=70e9;  
%E(2,:)=70e9;  
%E(3,:)=70e9;  
%E(4,:)=70e9;  
%E(5,:)=70e9;  
%E(6,:)=70e9;  
%E(7,:)=70e9;
```

%Vector que contiene el valor del área de cada barra

```
%A(1,:)=pi*(((1e-2)/2)^(2));  
%A(2,:)=pi*(((1e-2)/2)^(2));  
%A(3,:)=pi*(((1e-2)/2)^(2));  
%A(4,:)=pi*(((1e-2)/2)^(2));  
%A(5,:)=pi*(((1e-2)/2)^(2));  
%A(6,:)=pi*(((1e-2)/2)^(2));  
%A(7,:)=pi*(((1e-2)/2)^(2));
```

%vector que contiene el valor de los momentos de inercia de cada barra

```
%I(1,:)=pi*(((1e-2)/2)^(4))/2;  
%I(2,:)=pi*(((1e-2)/2)^(4))/2;  
%I(3,:)=pi*(((1e-2)/2)^(4))/2;  
%I(4,:)=pi*(((1e-2)/2)^(4))/2;  
%I(5,:)=pi*(((1e-2)/2)^(4))/2;  
%I(6,:)=pi*(((1e-2)/2)^(4))/2;  
%I(7,:)=pi*(((1e-2)/2)^(4))/2;
```

%Aqui se introducen los nodos de cada una de las barras

```
%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];
```

%En la variable SIZEBARRAS se guardará un vector que indicará la dimensión de la matriz BARRAS

```
SIZEBARRAS=size(BARRAS);
```

%Aqui se crea una matriz con las coordenadas de los nodos, en la que el

número de fila se corresponderá con el número del nodo. El primer valor (primera columna)

será la coordenada 'x' y el segundo (segunda columna) la coordenada 'y'.

```
%Nodo(1,:)= [0 0];  
%Nodo(2,:)= [0.25 0.433];  
%Nodo(3,:)= [0.5 0];  
%Nodo(4,:)= [0.75 0.433];  
%Nodo(5,:)= [1 0];
```

%La dimensión de la matriz Nodo, se memorizará en la variable 'dimensionkeestructura' para cálculos posteriores.

```
dimensionkeestructura=size(Nodo);
```

%La matriz RestriccionesNodo tendrá tres columnas y habrá que introducir

```
%las restricciones de cada nodo por filas, siendo el primer valor, la
restricción del
%desplazamiento horizontal, el segundo el vertical y el tercero el
giro.
%Si se trata de un grado restringido se indicara con un '0' y si por
el
%contrario es un grado de libertad no restringido, se indicará con un
'1'.
%Las condiciones de contorno vendrán determinadas por las
restricciones de
%los nodos, que habrá tantas como número de unos por dos tenga la
matriz.

%RestriccionesNodo(1,:)= [0 0 1]; %restricción del nodo 1 =
[desplazamiento horizontal nulo   desplazamiento vertical nulo
posibilidad de girar]
%RestriccionesNodo(2,:)= [1 1 1];
%RestriccionesNodo(3,:)= [1 1 1];
%RestriccionesNodo(4,:)= [1 1 1];
%RestriccionesNodo(5,:)= [0 0 1];

%La Matriz FuerzaNodos recoge la amplitud de la fuerza senoidal que se
va a
%aplicar, tiene 3 columnas, cada una se corresponde con los valores [x
y z],
%y el número de fila se corresponde con el número de nodo.

%Fuerzanodos(1,:)= [0 0 0];
%Fuerzanodos(2,:)= [0 0 0];
%Fuerzanodos(3,:)= [0 -1000 0];
%Fuerzanodos(4,:)= [0 0 0];
%Fuerzanodos(5,:)= [0 0 0];

%Bucle para obtener una matriz como la anterior, pero con los valores
de
%las fuerzas senoidales aplicadas a cada nodo.

for i=1:dimensionestructura(1,1)
FuerzasNodo(i,:)=FuerzaNodos(i,:);
end

%Aquí se creará un vector dispuesto en forma de columna, con tantas
filas
%como barras tenga la estructura y en la que cada número de fila se
%corresponderá con el número de la barra. Asignándose el valor de la
%constantes de rigidez en caso de que esa barra se considere un
muelle.

%Barramuelle(1,:)= [0];
%Barramuelle(2,:)= [0];
%Barramuelle(3,:)= [0];
%Barramuelle(4,:)= [0];
%Barramuelle(5,:)= [0];
%Barramuelle(6,:)= [0];
%Barramuelle(7,:)= [0];

%Mediante el siguiente bucle, se generará la matriz 'Barra', en el que
cada
%número de fila se corresponderá con el número de barra. Cada fila
tendrá 4 valores, los dos primeros serán
%las coordenadas del nodo menor de la barra y las otras dos las
coordenadas
```

```
%del nodo mayor.

for i=1:SIZEBARRAS(1,1)
    Barra(i,:)=[Nodo((BARRAS(i,1)),:) Nodo((BARRAS(i,2)),:)] ;
end

%La dimensión de la matriz 'Barra', se memorizar en la variable
%'barrassize' para cálculos posteriores.

barrassize=size(Barra);

%Bucle for e if con el que se calcula a partir de las coordenadas de
los nodos las
%longitudes de las barras, sus angulos

for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
    angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))

        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))

        alfa(i,:)=(2*pi)+angulo(i,:);
    end
end

%Bucle a partir del cual se calculan las matrices de rigidez de cada
barra en ejes globales agrupandolas
%en una matriz de 6 filas X n columnas. Posteriormente se utilizar la
%matriz 'k' para el cálculo de la matriz global de la estructura.

for i=1:barrassize(1,1)
    if (i==1) && (Barramuelle(i,1)==0)

k=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alfa(i,1)));
        elseif (i==1) && (Barramuelle(i,1)~=0)
            k=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        elseif i>1 && (Barramuelle(i,1)==0)
            k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alf
a(i,1)));
            elseif (i>1) && (Barramuelle(i,1)~=0)
                k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
```

```

    end

end

%Obtenci n de las matriz de ceros

for i=1:dimensionK
    if i==1
        CEROS=zeros(3);
    elseif i>1
        %filas
        CEROS((i*3)-2):(i*3), [1:(3*i)]=zeros(3,(i*3));
        %columnas
        CEROS(1:(i+(i-3)), [(3*i)-2:(3*i)]=zeros((i+(i-3)),3);
    end
end

%Bucle con el que se calcula la matriz de rigidez en ejes globales de
%nombre 'kglobales'

kglobales=CEROS;
for i=1:(barrassize(1,1))
    if (i==1)
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m])+k(1:3, [1:3]);
        %m,n
        kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n])+k(1:3, [4:6]);
        %n,m
        kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m])+k(4:6, [1:3]);
        %n,n
        kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n, [(2*n)+(n-2):3*n])+k(4:6, [4:6]);

    elseif i>1
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m]))+(k(1:3, [(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m]));
        %m,n
        kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n]))+(k(1:3, [(5*i)+(i-
2):6*i]))+(kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n]));
        %n,m
        kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m]))+(k(4:6, [(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m]));
        %n,n
        kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n, [(2*n)+(n-2):3*n]))+(k(4:6, [(5*i)+(i-
2):6*i]))+(kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n]));
    end
end

%Se le asigna la letra 'K' los valores que guarda la palabra

```

```
'kglobales'

K=kglobales;

%Estos dos bucles sirven para ordenar en forma de columna las
restricciones de los
% nodos de forma de forma que indique la posición del vector en el que
el
% grado no está restringido y mediante lo que se podrá obtener las
matrices
% reducidas para plantear el sistema de ecuaciones a resolver.

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif i==1
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==1)
            desplazamiento((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==2)
            desplazamiento((i*I)+(i-1),1)=((i*I)+(i-
1))*(RestriccionesNodo(i,I));
        elseif (i>1) && (I==3)
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        end
    end
end

sizedesplazamiento=size(desplazamiento);

%Se crea un único vector (el que iría multiplicado por la matriz que
contiene a
% la matriz de rigidez y de amortiguamiento) en el que la primera mitad
de valores se
% corresponden con los desplazamientos y la segunda con las velocidades

despvel=[desplazamiento];
sizedespvel=size(despvel);

%Bucle para obtener la posición de la primera incógnita de
desplazamientos
% y velocidades en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end

%Con esto se obtiene un vector columna con las posiciones de las
% velocidades y desplazamientos que son incógnita

dvr=despvelred;

for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end
```

```
end
end

sizedespvelred=size(despvelred);

%Para la obtencion de las matrices reducidas que contienen a la matriz
de
%rigidez y a la matriz de masas se emplea el siguiente bucle

for i=1:sizedespvelred(1,1)
    for I=1:sizedespvelred(1,1)
        if (i==1) && (I==1)
            H=[K(despvelred(i,1),despvelred(I,1))];

            elseif i>0
                H(i,I)=K(despvelred(i,1),despvelred(I,1));
            end
        end
    end

%Obtencion de la columna de las fuerzas aplicadas en los nodos

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            fuerza=FuerzasNodo(i,I);
        elseif i>-1
            fuerza=[fuerza;FuerzasNodo(i,I)];
        end
    end
end

sizefuerza=size(fuerza);

FUERZASCOLUMNA=[fuerza];

for i=1:sizedespvelred(1,1)
    if i==1
        fuerzared=FUERZASCOLUMNA(despvelred(i,1),1);
    elseif i>1
        fuerzared=[fuerzared;FUERZASCOLUMNA(despvelred(i,1),1)];
    end
end

%Obtención de desplazamientos de la estructura.

DesplazamientoXreticulada=H\fuerzared;
```

### Obtención de desplazamientos para estructuras reticuladas en dinámico.

```
function
dx=xprimareticuladaconcoordenadasymuellePRUEBA(t,x,O,H,tfin,FuerzaNodo
s,BARRAS,despvelred,omegaaa,desfase,dimensionkestructura)

%Vector desfase con tantas filas como nodos tenga la estructura, sera
ua de
%las componentes de la fuerza y habra que indicar en el caso de que en
u
%determinado nodo se aplique una fuerza senoidal, el valor de la
frecuencia
```



```
%desfase(1,:)=0;
%desfase(2,:)=0;
%desfase(3,:)=1000;
%desfase(4,:)=0;
%desfase(5,:)=0;

%Vector omega con tantas filas como nodos tenga la estructura, sera ua
de
%las componentes de la fuerza y habra que indicar en el caso de que en
u
%determinado nodo se aplique una fuerza senoidal, el valor de la
frecuencia

%omegaaa(1,:)=0;
%omegaaa(2,:)=0;
%omegaaa(3,:)=1000;
%omegaaa(4,:)=0;
%omegaaa(5,:)=0;

sizedespvelred=size(despvelred);

%Aqui se crea el vector 'FUERZA' en el que cada uno de los elementos
del
%vector ser el resultado de esta operacion
%"sin((omegaaa(i,:)*t)+desfase(i,:))"

for i=1:dimensionkestructura(1,1)
FUERZA(i,:)=sin((omegaaa(i,:)*t)+desfase(i,:));
end

%FuerzaNodos(1,:)= [0 0 0];
%FuerzaNodos(2,:)= [0 0 0];
%FuerzaNodos(3,:)= [0 -1 0];
%FuerzaNodos(4,:)= [0 0 0];
%FuerzaNodos(5,:)= [0 0 0];

%Bucle para obtener una matriz como la anterior, pero con los valores
de
%las fuerzas senoidales aplicadas a cada nodo.

for i=1:dimensionkestructura(1,1)
FuerzasNodo(i,:)=FuerzaNodos(i,:)*FUERZA(i,:);
end

%Obtencion de la columna de las fuerzas aplicadas en los nodos

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            fuerza=FuerzasNodo(i,I);
        elseif i>-1
            fuerza=[fuerza;FuerzasNodo(i,I)];
        end
    end
end

sizefuerza=size(fuerza);

cerosfuerza=zeros(sizefuerza(1,1),1);

%Aqui y se obtiene la culumna de fuerzas aplicadas a la estructura

FUERZASCOLUMNA=[cerosfuerza;fuerza];
```

```
%Aquí se obtiene la columna reducida de las fuerzas aplicadas a la
%estructura

for i=1:sizedespvelred(1,1)
    if i==1
        fuerzared=FUERZASCOLUMNA(despvelred(i,1),1);
    elseif i>1
        fuerzared=[fuerzared;FUERZASCOLUMNA(despvelred(i,1),1)];
    end
end

%Indica el porcentaje del proceso de cálculo

(t*100)/tfin

%Planteamiento de la ecuación a resolver por el ode45

dx=(O*fuerzared)-(O*(H*x));
```

## Anexo 5

En este anexo se encuentran los ficheros con los que se puede obtener las fuerzas y reacciones de estructuras articuladas y reticuladas sometidas a cargas estáticas y dinámicas.

### Obtención de fuerzas y reacciones para estructuras articuladas en estático.

```
function
ReaccionesyfuerzasReticulada=ReaccionesenestaticoestructuraReticulada(
E,A,I,BARRAS,Nodo,RestriccionesNodo,FuerzaNodos,Barramuelle)

Desplazamientosenestaticoestructurareticulada(E,A,I,BARRAS,Nodo,Restri
ccionesNodo,FuerzaNodos,Barramuelle);
desplazamientosenestructurareticulada=ans;
sizedesplazamientosenestructurareticulada=size(desplazamientosenestruc
turareticulada);

%TODO LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
%INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA

%Vector que contiene el valor del módulo de elasticidad de cada barra

%E(1,:)=70e9;
%E(2,:)=70e9;
%E(3,:)=70e9;
%E(4,:)=70e9;
%E(5,:)=70e9;
%E(6,:)=70e9;
%E(7,:)=70e9;

%Vector que contiene el valor del área de cada barra

%A(1,:)=pi*(((1e-2)/2)^(2));
%A(2,:)=pi*(((1e-2)/2)^(2));
%A(3,:)=pi*(((1e-2)/2)^(2));
```

```
%A(4,:)=pi*(((1e-2)/2)^(2));
%A(5,:)=pi*(((1e-2)/2)^(2));
%A(6,:)=pi*(((1e-2)/2)^(2));
%A(7,:)=pi*(((1e-2)/2)^(2));

%Aquí se introducen los nodos de cada de una de las barras

%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];

%En la variable SIZEBARRAS se guardar· un vector que indicar· la
dimensi·n de la matriz BARRAS

SIZEBARRAS=size(BARRAS);

%Aquí se crea una matriz con las coordenadas de los nodos, en la que
el
n·mero de fila se corresponder· con el n·mero del nodo. El primer
valor (primera columna)
ser· la coordenada 'x' y el segundo (segunda columna) la coordenada
'y'.

%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];

%La dimensi·n de la matriz Nodo, se memorizara en la variable
'dimensionkeestructura' para c·lculos posteriores.

dimensionkeestructura=size(Nodo);
dimensionK=dimensionkeestructura(1,1);

%La matriz RestriccionesNodo tendr· dos columnas y habr· que
introducir
%las restricciones de cada nodo por filas, siendo el primer valor, la
restricci·n del
%desplazamiento horizontal y el tercero el giro.
%Si se trata de un grado restringido se indicara con un '0' y si por
el
%contrario es un grado de libertad no restringido, se indicar· con un
'1'.
%Las condiciones de contorno vendr·n determinadas por las
restricciones de
%los nodos, que habr· tantas como n·mero de unos por dos tenga la
matriz.

%RestriccionesNodo(1,:)= [0 0]; %restriccion del nodo 1 =
[desplazamiento horizontal nulo    desplazamiento vertical nulo]
%RestriccionesNodo(2,:)= [1 1];
%RestriccionesNodo(3,:)= [1 1];
%RestriccionesNodo(4,:)= [1 1];
%RestriccionesNodo(5,:)= [0 0];

%Aquí se crear· un vector dispuesto en forma de columna, con tantas
filas
%como barras tenga la estructura y en en la que cada n·mero de fila se
%corresponder· con el n·mero de la barra. Assign·ndose el valor de la
%constantes de rigidez en caso de que esa barra se considere un
```

muelle.

```
%Barramuelle(1,:)=0;  
%Barramuelle(2,:)=0;  
%Barramuelle(3,:)=0;  
%Barramuelle(4,:)=0;  
%Barramuelle(5,:)=0;  
%Barramuelle(6,:)=0;  
%Barramuelle(7,:)=0;
```

%Mediante el siguiente bucle, se generar· la matriz 'Barra', en el que cada

n·mero de fila se corresponder· con el n·mero de barra. Cada fila tendr· 4 valores, los dos primeros ser·n

%las coordenadas del nodo menor de la barra y las otras dos las coordenadas

%del nodo mayor.

```
for i=1:SIZEBARRAS(1,1)  
    Barra(i,:)= [Nodo((BARRAS(i,1)),:), Nodo((BARRAS(i,2)),:)];  
end
```

%La dimensi·n de la matriz 'Barra', se memorizar· en la variable  
%'barrassize' para c·lculos posteriores.

```
barrassize=size(Barra);
```

%Bucle for e if con el que se calcula a partir de las coordenadas de los nodos las

%longitudes de las barras, sus angulos

```
for i=1:(barrassize(1,1))  
    a(i,1)=(Barra(i,3))-(Barra(i,1));  
    a(i,2)=(Barra(i,4))-(Barra(i,2));  
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);  
    angulo(i,:)=atan((a(i,2))/(a(i,1)));  
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))  
        alfa(i,:)=0;  
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))  
        alfa(i,:)=angulo(i,:);  
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))  
        alfa(i,:)=pi/2;  
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))  
        alfa(i,:)=(pi/2)+angulo(i,:);  
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))  
        alfa(i,:)=pi;  
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))  
        alfa(i,:)=pi+angulo(i,:);  
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))  
        alfa(i,:)=(3*pi)/2;  
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))  
        alfa(i,:)=(2*pi)+angulo(i,:);  
    end  
end
```

%Bucle a partir del cual se calculan las matrices de rigidez de cada barra en ejes globales agrupandolas

%en una matriz de 6 filas X n columnas. Posteriormente se utilizará la matriz 'k' para el cálculo de la matriz global de la estructura.

```
for i=1:(barrassize(1,1))
    if (i==1) && (Barramuelle(i,1)==0)

k=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alfa(i,1)));
        elseif (i==1) && (Barramuelle(i,1)~=0)
            k=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        elseif i>1 && (Barramuelle(i,1)==0)
            k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alf
a(i,1)));
        elseif (i>1) && (Barramuelle(i,1)~=0)
            k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
        end
end
```

%Obtención de las matriz de ceros

```
for i=1:dimensionK
    if i==1
        CEROS=zeros(3);

        elseif i>1
            %filas
            CEROS(((i*3)-2):(i*3), [1:(3*i)])=zeros(3,(i*3));
            %columnas
            CEROS(1:(i+(i-3)), (((3*i)-2):(3*i)))=zeros((i+(i-3)),3);
        end
end
```

%Bucle con el que se calcula la matriz de rigidez en ejes globales de nombre 'kglobales'

```
kglobales=CEROS;
for i=1:(barrassize(1,1))
    if (i==1)
        m=BARRAS(i,1);
        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m])+k(1:3, [1:3]);
        %m,n
        kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n])+k(1:3, [4:6]);
        %n,m
        kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m])+k(4:6, [1:3]);
        %n,n
        kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n, [(2*n)+(n-2):3*n])+k(4:6, [4:6]);

        elseif i>1
            m=BARRAS(i,1);
            n=BARRAS(i,2);
```

```

        %m,m
        kglobales((2*m)+(m-2):3*m,[(2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m,[(2*m)+(m-2):3*m]))+(k(1:3,[(4*i)+(i+(i-5)): (4*i)+(i+(i-
3))] ))+(kglobales((2*m)+(m-2):3*m,[(2*m)+(m-2):3*m]));
        %m,n
        kglobales((2*m)+(m-2):3*m,[(2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m,[(2*n)+(n-2):3*n]))+(k(1:3,[(5*i)+(i-
2):6*i]))+(kglobales((2*m)+(m-2):3*m,[(2*n)+(n-2):3*n]));
        %n,m
        kglobales((2*n)+(n-2):3*n,[(2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n,[(2*m)+(m-2):3*m]))+(k(4:6,[(4*i)+(i+(i-5)): (4*i)+(i+(i-
3))] ))+(kglobales((2*n)+(n-2):3*n,[(2*m)+(m-2):3*m]));
        %n,n
        kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n,[(2*n)+(n-2):3*n]))+(k(4:6,[(5*i)+(i-
2):6*i]))+(kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n]));

end

end

%Se le asigna la letra 'K' los valores que guarda la palabra
'kglobales'

K=kglobales;

%Se obtiene el vector de desplazamientos

for i=1:(sizedesplazamientosenestructurareticulada(1,2))
    despreti(:,i)=desplazamientosenestructurareticulada(:,i);
end

matrizdespvel=[despreti];

%Este bucle sirve para ordenar en forma de columna las restricciones
de los
%nodos de forma de forma que indique la posición del vector en el que
el
%grado no esta restringido y mediante lo que se podrá obtener las
matrices
%reducidas para plantear el sistema de ecuaciones a resolver.

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif i==1
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==1)
                desplazamiento((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==2)
                desplazamiento((i*I)+(i-1),1)=((i*I)+(i-
1))*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==3)

desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        end
    end
end

sizedesplazamiento=size(desplazamiento);

```

```
%Restricciondespvel nos aporta el vector columna total con los
%desplazamientos

despvel=[desplazamiento];

sizedespvel=size(despvel);

%Bucle para obtener la posicion de la primera incognita de
desplazamientos
%en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end

%Con esto se obtiene un vector columna con las posiciones de los
%desplazamientos que son incognita

dvr=despvelred;

for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end

sizedespvelred=size(despvelred);
sizematrizdespvel=size(matrizdespvel);

%Con este bucle se posicionan los desplazamientos de cada no en un su
%posici n en el vetor, para posteriormente obtenerse las reacciones

for i=1:sizedespvel(1,1)
    for I=1:sizematrizdespvel(1,1)
        if despvel(i,1)==0
            matcoldespvel(i,1)=0;
        elseif (despvel(i,1)>0) && (despvel(i,1)==despvelred(I,1))
            matcoldespvel(i,1)=matrizdespvel(I,1);
        end
    end
end

%Obtencion de las reacciones y fuerzas de la estructura

ReaccionesyfuerzasReticulada=K*matcoldespvel;
```

### Obtenci n de fuerzas y reacciones para estructuras articuladas en din mico.

```
function
rfuerza=reaccionesarticulconcoordenadasINTERFAZ(M,K,C,despvel,despvelr
ed,x)%E,A,densidad,x,BARRAS,Nodo,RestriccionesNodo,Barraamortiguador,B
arramuelle)
```

```
%Puesto que en la interfaz se ejecuta inmediatamente despues del
calculo de
%desplazamientos, cuyos resultados se guardan bajo la letra x. Se le
%asignan los valores de la letra x a la variable matrizx.

%[t,x]= ode45
(@xprimaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ,
[tini,tfin],condicionesiniciales,[],E,A,densidad,omegaaa,B,BARRAS,Nodo
,RestriccionesNodo,FuerzaNodos,Barramuelle,Barraamortiguador,tfin);

matrizx=x;

sizex=size(matrizx);

%Se obtiene una matriz en la que, cada columna, es el resultado de los
%desplazamientos de los nodos para un instante de tiempo

for i=1:(sizex(1,2))/2
    desp(:,i)=matrizx(:,i);
end

%Se obtiene una matriz en la que, cada columna, es el resultado de las
%velocidades de los nodos para un instante de tiempo

for i=((sizex(1,2))/2)+1:sizex(1,2)
    vel(:,(i-((sizex(1,2))/2)))=matrizx(:,i);
end

%Se obtiene una matriz en la que, cada columna, es el resultado de los
%aceleraciones de los nodos para un instante de tiempo, a partir de la
%derivada de la velocidad.

for i=((sizex(1,2))/2)+1:sizex(1,2)
    ace(:,(i-(sizex(1,2))/2))=diff((vel(:,(i-(sizex(1,2))/2)))));
    %ace(:,(i-(sizex(1,2))/2))=[ace(:,(i-
(sizex(1,2))/2));ace(((sizex(1,1))-1),(i-((sizex(1,2))/2)))]];
end

ace=[ace;zeros(1,(sizex(1,2)/2))];
cerosace=zeros(sizex(1,1),sizex(1,2)/2);

for i=1:sizex(1,2)/2
    cerosace(sizex(1,1),i)=cerosace(sizex(1,1),i)+ace(sizex(1,1)-1,i);
end

ace=cerosace+ace;

%Matriz que contiene los datos de desplazamientos y velocidades

matrizdespvel=[desp vel];

%Matriz que contiene los datos de velocidades y aceleraciones

matrizvelace=[vel ace];

sizedespvel=size(despvel);
sizedespvelred=size(despvelred);
sizematrizdespvel=size(matrizdespvel);
sizematrizvelace=size(matrizvelace);

%Con este bucle se obtiene la matriz deslpzamientos y velocidades para
```



```
los distintos
%tiempos, que sera necesaria a la hora de multiplicarlo por la matriz
h que
%contiene la matriz de rigidez

for i=1:sizematrizdespvel(1,1)
for I=1:sizedespvel(1,1)
for QQ=1:sizedespvelred(1,1)
if despvel(I,1)==0
matcoldespvel(I,i)=0;
elseif (despvel(I,1)>0) && (despvel(I,1)==despvelred(QQ,1))
matcoldespvel(I,i)=matrizdespvel(i,QQ);
end
end
end
end

%Para obtener la matriz solo con los desplazamientos, que ira
multiplicado
%por la matriz de rigidez

sizematcoldespvel=size(matcoldespvel);

for i=1:(sizematcoldespvel(1,1))/2
    matcoldespvelred(i,:)=matcoldespvel(i,:);
end

%Para obtener la matriz solo con las velocidades que ira multiplicado
por
%la matriz de amortiguamientos

for i=((sizematcoldespvel(1,1))/2)+1:(sizematcoldespvel(1,1))
    matcolveldespred((i-
((sizematcoldespvel(1,1))/2)),:)=matcoldespvel(i,:);
    (i*100)/(sizematcoldespvel(1,1))
end

%Con este bucle se obtiene la matriz velocidades y acelereaciones para
los distintos
%tiempos, que sera necesaria a la hora de multiplicarlo por la matriz
g que
%contiene la matriz de masas

for i=1:sizematrizvelace(1,1)
for I=1:sizedespvel(1,1)
for QQ=1:sizedespvelred(1,1)
if despvel(I,1)==0
matcolvelace(I,i)=0;
elseif (despvel(I,1)>0) && (despvel(I,1)==despvelred(QQ,1))
matcolvelace(I,i)=matrizvelace(i,QQ);
end
end
end
end

%Para obtener la matriz solo con las acelereaciones que ira
multiplicado por
%la matriz de masas

for i=((sizematcoldespvel(1,1))/2)+1:(sizematcoldespvel(1,1))
    matcolvelacered((i-
((sizematcoldespvel(1,1))/2)),:)=matcolvelace(i,:);
```

```
end

%Si se quiere añadir la barra de proceso, para ver el progreso de
los
%calculos en pantalla, quitar el tanto por ciento a la waitbar

%wait=waitbar(0,'Sigue esperando');

%En este bucle se calculan las reacciones y fuerzas de la estructura
for i=1:(size(x,1))

    velamort=matcolvelpred(:,i);
    deveac=matcolvelred(:,i);
    xprima=matcolvelacered(:,i);
    rfuerza(:,i)=(M*xprima)+(K*deveac)+(C*velamort);
    %waitbar(i/(size(x,1)),(i*100)/(size(x,1)));

end
rfuerza(size(x,1)/2,(size(x,1)));
%close(wait);
```

### Obtención de fuerzas y reacciones en estructuras reticuladas en estático

```
function
ReaccionesyfuerzasReticulada=ReaccionesenestaticoestructuraReticulada(
E,A,I,BARRAS,Nodo,RestriccionesNodo,FuerzaNodos,Barramuelle)

Desplazamientosenestaticoestructurareticulada(E,A,I,BARRAS,Nodo,Restri
ccionesNodo,FuerzaNodos,Barramuelle);
desplazamientosenestaticoestructurareticulada=ans;
sizedesplazamientosenestaticoestructurareticulada=size(desplazamientosenestatic
oestructurareticulada);

% TODOS LOS DATOS QUE ESTAN EN VERDE, SON DATOS QUE SUPUESTAMENTE DEBE
% INTRODUCIR EL USUARIO Y QUE POR TANTO AHORA MISMO NO SERVIRIAN PARA
NADA

%Vector que contiene el valor del módulo de elasticidad de cada barra

%E(1,:)=70e9;
%E(2,:)=70e9;
%E(3,:)=70e9;
%E(4,:)=70e9;
%E(5,:)=70e9;
%E(6,:)=70e9;
%E(7,:)=70e9;

%Vector que contiene el valor del área de cada barra

%A(1,:)=pi*(((1e-2)/2)^(2));
%A(2,:)=pi*(((1e-2)/2)^(2));
%A(3,:)=pi*(((1e-2)/2)^(2));
%A(4,:)=pi*(((1e-2)/2)^(2));
%A(5,:)=pi*(((1e-2)/2)^(2));
%A(6,:)=pi*(((1e-2)/2)^(2));
%A(7,:)=pi*(((1e-2)/2)^(2));

%vector que contiene el valor de los momentos de inercia de cada
barra
```

```
%I(1,:)=pi*(((1e-2)/2)^(4))/2);
%I(2,:)=pi*(((1e-2)/2)^(4))/2);
%I(3,:)=pi*(((1e-2)/2)^(4))/2);
%I(4,:)=pi*(((1e-2)/2)^(4))/2);
%I(5,:)=pi*(((1e-2)/2)^(4))/2);
%I(6,:)=pi*(((1e-2)/2)^(4))/2);
%I(7,:)=pi*(((1e-2)/2)^(4))/2);

%Aquí se introducen los nodos de cada una de las barras

%BARRAS=[1 2;1 3;2 3;2 4;3 4;3 5;4 5];

%En la variable SIZEBARRAS se guardará un vector que indicará la
dimensión de la matriz BARRAS

SIZEBARRAS=size(BARRAS);

%Aquí se crea una matriz con las coordenadas de los nodos, en la que
el
número de fila se corresponderá con el número del nodo. El primer
valor (primera columna)
será la coordenada 'x' y el segundo (segunda columna) la coordenada
'y'.

%Nodo(1,:)= [0 0];
%Nodo(2,:)= [0.25 0.433];
%Nodo(3,:)= [0.5 0];
%Nodo(4,:)= [0.75 0.433];
%Nodo(5,:)= [1 0];

%La dimensión de la matriz Nodo, se memorizará en la variable
'dimensionkeestructura' para cálculos posteriores.

dimensionkeestructura=size(Nodo);
dimensionK=dimensionkeestructura(1,1);

%La matriz RestriccionesNodo tendrá tres columnas y habrá que
introducir
las restricciones de cada nodo por filas, siendo el primer valor, la
restricción del
desplazamiento horizontal, el segundo el vertical y el tercero el
giro.
%Si se trata de un grado restringido se indicará con un '0' y si por
el
contrario es un grado de libertad no restringido, se indicará con un
'1'.
%Las condiciones de contorno vendrán determinadas por las
restricciones de
los nodos, que habrá tantas como número de unos por dos tenga la
matriz.

%RestriccionesNodo(1,:)= [0 0 1]; %restriccion del nodo 1 =
[desplazamiento horizontal nulo desplazamiento vertical nulo
posibilidad de girar]
%RestriccionesNodo(2,:)= [1 1 1];
%RestriccionesNodo(3,:)= [1 1 1];
%RestriccionesNodo(4,:)= [1 1 1];
%RestriccionesNodo(5,:)= [0 0 1];

%Aquí se creará un vector dispuesto en forma de columna, con tantas
```

```
filas
%como barras tenga la estructura y en en la que cada número de fila se
%corresponder con el número de la barra. Asignándose el valor de la
%constantes de rigidez en caso de que esa barra se considere un
muelle.

%Barramuelle(1,:)=0;
%Barramuelle(2,:)=0;
%Barramuelle(3,:)=0;
%Barramuelle(4,:)=0;
%Barramuelle(5,:)=0;
%Barramuelle(6,:)=0;
%Barramuelle(7,:)=0;

%Mediante el siguiente bucle, se generará la matriz 'Barra', en el que
cada
número de fila se corresponder con el número de barra. Cada fila
tendrá 4 valores, los dos primeros serán
%las coordenadas del nodo menor de la barra y las otras dos las
coordenadas
%del nodo mayor.

for i=1:SIZEBARRAS(1,1)
    Barra(i,:)=[Nodo((BARRAS(i,1)),:) Nodo((BARRAS(i,2)),:)] ;
end

%La dimensión de la matriz 'Barra', se memorizará en la variable
%'barrassize' para cálculos posteriores.

barrassize=size(Barra);

%Bucle for e if con el que se calcula a partir de las coordenadas de
los nodos las
%longitudes de las barras, sus ángulos

for i=1:(barrassize(1,1))
    a(i,1)=(Barra(i,3))-(Barra(i,1));
    a(i,2)=(Barra(i,4))-(Barra(i,2));
    L(1,i)=(((a(i,1))^2)+((a(i,2))^2))^(0.5);
    angulo(i,:)=atan((a(i,2))/(a(i,1)));
    if ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=0;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=pi/2;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))<(Barra(i,4)))
        alfa(i,:)=(pi/2)+angulo(i,:);
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))==(Barra(i,4)))
        alfa(i,:)=pi;
    elseif ((Barra(i,1))>(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=pi+angulo(i,:);
    elseif ((Barra(i,1))==(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(3*pi)/2;
    elseif ((Barra(i,1))<(Barra(i,3))) && ((Barra(i,2))>(Barra(i,4)))
        alfa(i,:)=(2*pi)+angulo(i,:);
```

```

        end
    end

    %Bucle a partir del cual se calculan las matrices de rigidez de cada
    barra en ejes globales agrupandolas
    %en una matriz de 6 filas X n columnas. Posteriormente se utilizará la
    %matriz 'k' para el cálculo de la matriz global de la estructura.

    for i=1:(barrassize(1,1))
        if (i==1) && (Barramuelle(i,1)==0)

            k=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alfa(i,1)));
            elseif (i==1) && (Barramuelle(i,1)~=0)
                k=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
            elseif i>1 && (Barramuelle(i,1)==0)
                k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesreticuladasb(E(i,1),A(i,1),(L(1,i)),I(i,1),(alf
a(i,1)));
            elseif (i>1) && (Barramuelle(i,1)~=0)
                k(1:6, [(2*i)*3]-
5:((2*i)*3))=kglobalesmuellereticuladas(Barramuelle(i,1),alfa(i,1));
            end
        end

    end

    %Obtención de las matriz de ceros

    for i=1:dimensionK
        if i==1
            CEROS=zeros(3);
        elseif i>1
            %filas
            CEROS((i*3)-2:(i*3), [1:(3*i)])=zeros(3,(i*3));
            %columnas
            CEROS(1:(i+(i-3)), [(3*i)-2:(3*i)])=zeros((i+(i-3)),3);
        end
    end

    %Bucle con el que se calcula la matriz de rigidez en ejes globales de
    %nombre 'kglobales'

    kglobales=CEROS;
    for i=1:(barrassize(1,1))
        if (i==1)
            m=BARRAS(i,1);
            n=BARRAS(i,2);
            %m,m
            kglobales((2*m)+(m-2):3*m, [(2*m)+(m-2):3*m])=CEROS((2*m)+(m-
2):3*m, [(2*m)+(m-2):3*m])+k(1:3, [1:3]);
            %m,n
            kglobales((2*m)+(m-2):3*m, [(2*n)+(n-2):3*n])=CEROS((2*m)+(m-
2):3*m, [(2*n)+(n-2):3*n])+k(1:3, [4:6]);
            %n,m
            kglobales((2*n)+(n-2):3*n, [(2*m)+(m-2):3*m])=CEROS((2*n)+(n-
2):3*n, [(2*m)+(m-2):3*m])+k(4:6, [1:3]);
            %n,n
            kglobales((2*n)+(n-2):3*n, [(2*n)+(n-2):3*n])=CEROS((2*n)+(n-
2):3*n, [(2*n)+(n-2):3*n])+k(4:6, [4:6]);

            elseif i>1
                m=BARRAS(i,1);

```

```

        n=BARRAS(i,2);
        %m,m
        kglobales((2*m)+(m-2):3*m,[(2*m)+(m-2):3*m])=(CEROS((2*m)+(m-
2):3*m,[(2*m)+(m-2):3*m]))+(k(1:3,[(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*m)+(m-2):3*m,[(2*m)+(m-2):3*m]));
        %m,n
        kglobales((2*m)+(m-2):3*m,[(2*n)+(n-2):3*n])=(CEROS((2*m)+(m-
2):3*m,[(2*n)+(n-2):3*n]))+(k(1:3,[(5*i)+(i-
2):6*i]))+(kglobales((2*m)+(m-2):3*m,[(2*n)+(n-2):3*n]));
        %n,m
        kglobales((2*n)+(n-2):3*n,[(2*m)+(m-2):3*m])=(CEROS((2*n)+(n-
2):3*n,[(2*m)+(m-2):3*m]))+(k(4:6,[(4*i)+(i+(i-5)):(4*i)+(i+(i-
3))]))+(kglobales((2*n)+(n-2):3*n,[(2*m)+(m-2):3*m]));
        %n,n
        kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n])=(CEROS((2*n)+(n-
2):3*n,[(2*n)+(n-2):3*n]))+(k(4:6,[(5*i)+(i-
2):6*i]))+(kglobales((2*n)+(n-2):3*n,[(2*n)+(n-2):3*n]));

end

end

%Se le asigna la letra 'K' los valores que guarda la palabra
'kglobales'

K=kglobales;

%Se obtiene el vector de desplazamientos

for i=1:(sizedesplazamientosenestructurareticulada(1,2))
    despreti(:,i)=desplazamientosenestructurareticulada(:,i);
end

matrizdespvel=[despreti];

%Este bucle sirve para ordenar en forma de columna las restricciones
de los
%nodos de forma de forma que indique la posici n del vector en el que
el
%grado no esta restringido y mediante lo que se podr  obtener las
matrices
%reducidas para plantear el sistema de ecuaciones a resolver.

for i=1:max(max(BARRAS))
    for I=1:3
        if (i==1) && (I==1)
            desplazamiento=i*I*RestriccionesNodo(i,I);
        elseif i==1
            desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==1)
                desplazamiento((i*I)+(i+(i-2)),1)=((i*I)+(i+(i-
2)))*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==2)
                desplazamiento((i*I)+(i-1),1)=((i*I)+(i-
1))*(RestriccionesNodo(i,I));
            elseif (i>1) && (I==3)

desplazamiento(i*I,1)=i*I*(RestriccionesNodo(i,I));
        end
    end
end

```

```
sizedesplazamiento=size(desplazamiento);

%Restricciondespvel nos aporta el vector columna total con los
%desplazamientos

despvel=[desplazamiento];

sizedespvel=size(despvel);

%Bucle para obtener la posicion de la primera incognita de
desplazamientos
%en la reducida

despvelred=-1;
for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvelred==-1)
        despvelred=[despvel(i,1)];
    end
end

%Con esto se obtiene un vector columna con las posiciones de los
desplazamientos que son incognita

dvr=despvelred;

for i=1:sizedespvel(1,1)
    if (despvel(i,1)~=0) && (despvel(i,1)>dvr)
        despvelred=[despvelred;despvel(i,1)];
        dvr=despvel(i,1);
    end
end

sizedespvelred=size(despvelred);
sizematrizdespvel=size(matrizdespvel);

%Con este bucle se posicionan los desplazamientos de cada no en un su
%posici n en el vetor, para posteriormente obtenerse las reacciones

for i=1:sizedespvel(1,1)
for I=1:sizematrizdespvel(1,1)
if despvel(i,1)==0
matcoldespvel(i,1)=0;
elseif (despvel(i,1)>0) && (despvel(i,1)==despvelred(I,1))
matcoldespvel(i,1)=matrizdespvel(I,1);
end
end
end

%obtencion de las reacciones y fuerzas de la estructura

ReaccionesyfuerzasReticulada=K*matcoldespvel;
```

### Obtenci n de fuerzas y reacciones en estructuras reticuladas en din mico

```
function
rfuerza=reaccionesreticulconcoordenadasINTERFAZ(M,K,C,despvel,despvelr
```

```
ed,x)

%Puesto que en la interfaz se ejecuta inmediatamente despues del
calculo de
%desplazamientos, cuyos resultados se guardan bajo la letra x. Se le
%asignan los valores de la letra x a la variable matrizx.

%[t,x]= ode45 (@xprimareticuladaconcoordenadas, [0,20e-
3*pi],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]);
matrizx=x;

sizex=size(matrizx);

%Se obtiene una matriz en la que, cada columna, es el resultado de los
%desplazamientos de los nodos para un instante de tiempo

for i=1:(sizex(1,2))/2
    desp(:,i)=matrizx(:,i);
    (i*100)/(sizex(1,2))/2
end
7

%Se obtiene una matriz en la que, cada columna, es el resultado de las
%velocidades de los nodos para un instante de tiempo

for i=((sizex(1,2))/2)+1:sizex(1,2)
    vel(:,(i-((sizex(1,2))/2)))=matrizx(:,i);
    (i*100)/sizex(1,2)
end
6

%Se obtiene una matriz en la que, cada columna, es el resultado de los
%aceleraciones de los nodos para un instante de tiempo, a partir de la
%derivada de la velocidad.

for i=((sizex(1,2))/2)+1:sizex(1,2)
    ace(:,(i-(sizex(1,2))/2))=diff((vel(:,(i-(sizex(1,2))/2)))));
    (i*100)/sizex(1,2)
end
5
ace=[ace;zeros(1,(sizex(1,2))/2)];
cerosace=zeros(sizex(1,1),sizex(1,2)/2);

for i=1:sizex(1,2)/2
    cerosace(sizex(1,1),i)=cerosace(sizex(1,1),i)+ace(sizex(1,1)-1,i);
    (i*100)/(sizex(1,2)/2)
end
4
ace=cerosace+ace;

%Matriz que contiene los datos de desplazamientos y velocidades

matrizdespvel=[desp vel];

%Matriz que contiene los datos de velocidades y aceleraciones

matrizvelace=[vel ace];

sizedespvel=size(despvel);
sizedespvelred=size(despvelred);
sizematrizdespvel=size(matrizdespvel);
```



```
sizematrizvelace=size(matrizvelace);

%Con este bucle se obtiene la matriz desplazamientos y velocidades para
los distintos
%tiempos, que sera necesaria a la hora de multiplicarlo por la matriz
h que
%contiene la matriz de rigidez

for i=1:sizematrizdespvel(1,1)
for I=1:sizedespvel(1,1)
for QQ=1:sizedespvelred(1,1)
if despvel(I,1)==0
matcoldespvel(I,i)=0;
elseif (despvel(I,1)>0) && (despvel(I,1)==despvelred(QQ,1))
matcoldespvel(I,i)=matrizdespvel(i,QQ);
end
end
end
end

%Para obtener la matriz solo con los desplazamientos, que ira
multiplicado
%por la matriz de rigidez

sizematcoldespvel=size(matcoldespvel);

for i=1:(sizematcoldespvel(1,1))/2
matcoldespvelred(i,:)=matcoldespvel(i,:);
(i*100)/(sizematcoldespvel(1,1))/2
end
3

%Para obtener la matriz solo con las velocidades que ira multiplicado
por
%la matriz de amortiguamientos

for i=((sizematcoldespvel(1,1))/2)+1:(sizematcoldespvel(1,1))
matcolveldespred((i-
((sizematcoldespvel(1,1))/2)),:)=matcoldespvel(i,:);
(i*100)/(sizematcoldespvel(1,1))
end
2

%Con este bucle se obtiene la matriz velocidades y acelereaciones para
los distintos
%tiempos, que sera necesaria a la hora de multiplicarlo por la matriz
g que
%contiene la matriz de masas

for i=1:sizematrizvelace(1,1)
for I=1:sizedespvel(1,1)
for QQ=1:sizedespvelred(1,1)
if despvel(I,1)==0
matcolvelace(I,i)=0;
elseif (despvel(I,1)>0) && (despvel(I,1)==despvelred(QQ,1))
matcolvelace(I,i)=matrizvelace(i,QQ);
end
end
end
end

%Para obtener la matriz solo con las acelereaciones que ira
multiplicado por
```

```
%la matriz de masas

for i=((sizematcoldespvel(1,1))/2)+1:(sizematcoldespvel(1,1))
    matcolvelacered((i-
((sizematcoldespvel(1,1))/2)),:)=matcolvelace(i,:);
    (i*100)/(sizematcoldespvel(1,1))
end
1

%Si se quiere añadir la barra de proceso, para ver el progreso de
los
%calculos en pantalla, quitar el tanto por ciento a la waitbar

%wait=waitbar(0,'Sigue esperando');

%En este bucle se calculan las reacciones y fuerzas de la estructura

for i=1:(sizex(1,1))
    velamort=matcolveldespred(:,i);
    deveac=matcoldespvelred(:,i);
    xprima=matcolvelacered(:,i);
    rfuerza(:,i)=(M*xprima)+(K*deveac)+(C*velamort);
    %waitbar(i/(sizex(1,1)));
    (i*100)/(sizex(1,1))
end
rfuerza(sizedespvel(1,1)/2,(sizex(1,1)));
%close(wait);
```

## Anexo 6

En este anexo se encuentran todos los ficheros relacionados con la programación del interfaz.

### Menú principal

```
function varargout = Menu_Principal(varargin)
% MENU_PRINCIPAL M-file for Menu_Principal.fig
%     MENU_PRINCIPAL, by itself, creates a new MENU_PRINCIPAL or
%     raises the existing
%     singleton*.
%
%     H = MENU_PRINCIPAL returns the handle to a new MENU_PRINCIPAL
%     or the handle to
%     the existing singleton*.
%
%     MENU_PRINCIPAL('CALLBACK',hObject,eventData,handles,...) calls
%     the local
%     function named CALLBACK in MENU_PRINCIPAL.M with the given
%     input arguments.
%
%     MENU_PRINCIPAL('Property','Value',...) creates a new
%     MENU_PRINCIPAL or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before Menu_Principal_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to Menu_Principal_OpeningFcn via
%     varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Menu_Principal

% Last Modified by GUIDE v2.5 20-Nov-2011 18:54:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Menu_Principal_OpeningFcn, ...
                  'gui_OutputFcn',  @Menu_Principal_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT

% --- Executes just before Menu_Principal is made visible.
function Menu_Principal_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Menu_Principal (see VARARGIN)

% Choose default command line output for Menu_Principal
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Menu_Principal wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Menu_Principal_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in desparti_menu.
function desparti_menu_Callback(hObject, eventdata, handles)
% hObject      handle to desparti_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Abre la ventana que permite definir las características de una
estructura articulada
Definicion_Estructura_Articulada
%Espera hasta que la ventana de Definicion_Estructura_Articulada se
cierre
uiwait

% --- Executes on button press in fuerzreti_menu.
function fuerzreti_menu_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzreti_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Abre la ventana FuerzasReticulada, en la que se comprueban los
resultados
%de las reacciones y fuerzas en un estructura reticulada sometida a
cargas
%dinamicas
FuerzasReticulada

uiwait

% --- Executes on button press in despreti_menu.
```

```
function despreti_menu_Callback(hObject, eventdata, handles)
% hObject      handle to despreti_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Abre la ventana que permite definir las características de una
estructura
%articulada
Definicion_Estructura_Reticulada

uiwait

% --- Executes on button press in fuerzarti_menu.
function fuerzarti_menu_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzarti_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Abre la ventana FuerzasReticulada, en la que se comprueban los
resultados
%de las reacciones y fuerzas en un estructura reticulada sometida a
cargas
%dinamicas
FuerzasArticuladas

uiwait

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se limpian todos los datos introducidos y obtenidos, ademas de, la
ventana de comandos dejandola en blanco
% posteriormente se cierra la ventana Menu_Principal

clear all
clc
close Menu_Principal

% --- Executes on button press in desplazamientoarticualdas_menuppal.
function desplazamientoarticualdas_menuppal_Callback(hObject,
eventdata, handles)
% hObject      handle to desplazamientoarticualdas_menuppal (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre una ventana en la que se pueden comprobar los resultados de
los
%desplazamientos para estructuras articuladas en dinamico
Desplazamiento_Articualdas

uiwait

% --- Executes on button press in desplazamientosreticulada_menuppal.
function desplazamientosreticulada_menuppal_Callback(hObject,
eventdata, handles)
```

```
% hObject    handle to desplazamientosreticulada_menuppal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se abre una ventana en la que se pueden comprobar los resultados de
los
%desplazamientos para estructuras reticuladas en dinamico
Desplazamiento_Reticuladas

uiwait

% --- Executes on button press in fuerzasestaticoA_menuppal.
function fuerzasestaticoA_menuppal_Callback(hObject, eventdata,
handles)
% hObject    handle to fuerzasestaticoA_menuppal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se abre una ventana en la que se definen las caractertisticas de las
%fuerzas aplicadas sobre una estructura sometida a cargas estaticas
Fuerza_Estatico_Articulada

uiwait

% --- Executes on button press in fuerzasdinamicoA_menuppal.
function fuerzasdinamicoA_menuppal_Callback(hObject, eventdata,
handles)
% hObject    handle to fuerzasdinamicoA_menuppal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se abre una ventana en la que se definen las caractertisticas de las
%fuerzas aplicadas sobre una estructura sometida a cargas dinamicas
Fuerza_Dinamica_Articulada

uiwait

% --- Executes on button press in valorespropiosA_menuppal.
function valorespropiosA_menuppal_Callback(hObject, eventdata,
handles)

%se declaran las variables globales que se van a necesitar para poder
%calcular los valores propios de la estructura articulada
global E A densidad BARRAS Nodo RestriccionesNodo Barramuelle
ValorespropiosArticulada ans

%Mediante la funcion valores_propios_estructura_articulada.m y a
partir de
%los datos que vienen entre parentesis, se produce el calculo de los
%valores y modos propios de una estructura articulada
valores_propios_estructura_articulada(E,A,densidad,BARRAS,Nodo,Restric
cionesNodo,Barramuelle);

%El resultado obtenido de la funcion
%valores_propios_estructura_articulada.m, se guarda en la palabra
%ValorespropiosArticulada
ValorespropiosArticulada=ans;

%Se abre la ventana donde se muestran los resultados y se espera hasta
que
%el usuario la cierre.
Valores_Propios_EA
```

```
uiwait

% --- Executes on button press in desplazamientoestaticoR_menuppal.
function desplazamientoestaticoR_menuppal_Callback(hObject, eventdata, handles)

%Se abre una ventana en la que se muestran los resultados de los
%desplazamientos en una estructura reticulada sometida a cargas
%estaticas
Calculo_Desplazamientos_ER_Estatico

%se espera hasta que el usuario cierra esa ventana.
uiwait

% --- Executes on button press in reaccionesestaticoR_menuppal.
function reaccionesestaticoR_menuppal_Callback(hObject, eventdata, handles)

%Se abre una ventana en la que se muestran los resultados de las
fuerzas y
%reacciones obtenidas al someter a una estructura reticulada a cargas
%estaticas
Calculo_Fuerzas_ER_Estatico

uiwait

% --- Executes on button press in desplazamientoestaticoA_menuppal.
function desplazamientoestaticoA_menuppal_Callback(hObject, eventdata, handles)

%Se abre una ventana en la que se muestran los resultados de los
%desplazamientos al someter una estructura articulada a cargas
%estaticas
Calculo_Desplazamientos_EA_Estatico

%Se espera a volver a la ventana de Menu_principal, hasta que el
usuario
%cierre la ventana anteriormente mencionada
uiwait

% --- Executes on button press in reaccionesestaticoA_menuppal.
function reaccionesestaticoA_menuppal_Callback(hObject, eventdata, handles)

%Se abre una ventana en la que se muestran los resultados de las
fuerzas y
%reacciones obtenidas al someter una estructura articulada a cargas
%estaticas
Calculo_Fuerzas_EA_Estatico
%Se espera a volver a la ventana de Menu_principal, hasta que el
usuario
%cierre la ventana anteriormente mencionada
uiwait

% --- Executes on button press in fuerzasdinamicoR_menuppal.
function fuerzasdinamicoR_menuppal_Callback(hObject, eventdata, handles)

%Se abre una ventana que permite definir las características de las
fuerzas
%dinamicas que se van a aplicar sobre la estructura reticulada
```

```
Fuerza_Dinamica_Reticulada
%Se espera a volver a la ventana de Menu_principal, hasta que el
usuario
%cierre la ventana anteriormente mencionada
uiwait
% --- Executes on button press in valorespropiosR_menuppal.
function valorespropiosR_menuppal_Callback(hObject, eventdata,
handles)

%Variables globales
global E A I densidad BARRAS Nodo RestriccionesNodo Barramuelle
ValorespropiosReticulada ans
%Se toman los valores de las variables globales para introducirlas en
la
%funcion valores_propios_estructura_reticulada.m y asi poder obtener
los
%valores y modos propios de una estructura reticulada
valores_propios_estructura_reticulada(E,A,I,densidad,Nodo,BARRAS,Restr
iccionesNodo,Barramuelle)
%Se recogen en ValorespropiosReticulada los valores y modos propios de
la
%estructura reticulada
ValorespropiosReticulada=ans;
%Se abre una ventana en la que se muestran los resultados de los
valores y
%modos propios de la estructura reticulada
Valores_Propios_ER
%Se espera a volver a la ventana de Menu_principal, hasta que el
usuario
%cierre la ventana anteriormente mencionada
uiwait

% --- Executes on button press in fuerzasestaticoR_menuppal.
function fuerzasestaticoR_menuppal_Callback(hObject, eventdata,
handles)

%Se abre una ventana en la que se pueden definir las caracterisitas
de las
%fuerzas estaticas aplicadas sobre una estructura reticulada
Fuerza_Estatico_Reticulada
%Se espera a volver a la ventana de Menu_principal, hasta que el
usuario
%cierre la ventana anteriormente mencionada
uiwait
```

### Definición estructura articulada

```
function varargout = Definicion_Estructura_Articulada(varargin)
% DEFINICION_ESTRUCTURA_ARTICULADA M-file for
Definicion_Estructura_Articulada.fig
%     DEFINICION_ESTRUCTURA_ARTICULADA, by itself, creates a new
DEFINICION_ESTRUCTURA_ARTICULADA or raises the existing
%     singleton*.
%
%     H = DEFINICION_ESTRUCTURA_ARTICULADA returns the handle to a
new DEFINICION_ESTRUCTURA_ARTICULADA or the handle to
%     the existing singleton*.
%
%
DEFINICION_ESTRUCTURA_ARTICULADA('CALLBACK',hObject,eventData,handles,
...) calls the local
%     function named CALLBACK in DEFINICION_ESTRUCTURA_ARTICULADA.M
```



```
with the given input arguments.
%
%     DEFINICION_ESTRUCTURA_ARTICULADA('Property','Value',...)
creates a new DEFINICION_ESTRUCTURA_ARTICULADA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Definicion_Estructura_Articulada_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Definicion_Estructura_Articulada_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools
definicion_estructura_articulada. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Definicion_Estructura_Articulada

% Last Modified by GUIDE v2.5 20-Nov-2011 12:38:30

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Definicion_Estructura_Articulada_OpeningFcn, ...
                  'gui_OutputFcn',   @Definicion_Estructura_Articulada_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Definicion_Estructura_Articulada is made
visible.
function Definicion_Estructura_Articulada_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to
Definicion_Estructura_Articulada (see VARARGIN)

% Choose default command line output for
Definicion_Estructura_Articulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes Definicion_Estructura_Articulada wait for user response
(see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout =
Definicion_Estructura_Articulada_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%Variables globales
global numero_nodos tablabarras tablactek tablacoordenadasnodos
tablarestricciones tablactec tablasecciones numero_barras
numero_restricciones numero_muelles numero_amortiguadores
%Se muestra el valor de las variables en la ventana de Definición
%Estructura Articulada
set(handles.nodo_output,'String',numero_nodos)
set(handles.barras_output,'data',tablabarras)
set(handles.valorctek_output,'data',tablactek)
set(handles.coordenadasnodos_output,'data',tablacoordenadasnodos)
set(handles.desplazamientos_output,'data',tablarestricciones)
set(handles.valorctec_output,'data',tablactec)
set(handles.tablasecciones_output,'data',tablasecciones)
set(handles.barra_output,'String',numero_barras)
set(handles.restricci_output,'String',numero_restricciones)
set(handles.nummuelles_output,'String',numero_muelles)
set(handles.numamortiguadores_output,'String',numero_amortiguadores)

% --- Executes on button press in numnodos_menu.
function numnodos_menu_Callback(hObject, eventdata, handles)

%Se abre una ventana en la que habra que introducir el numero de nodos
de la
%estructura articulada
Numero_Nodos_EA

uiwait

%se muestra el valor de la variable numero_nodos en la caja de edicion
de
%texto no editable de la ventana Definicion Estructura Articulada
global numero_nodos
set(handles.nodo_output,'String',numero_nodos)

% --- Executes on button press in barras_menu.
function barras_menu_Callback(hObject, eventdata, handles)

global numero_barras BARRAS  contadornumerobarras tablabarras

%Mediante el siguiente bucles ejecutado en la ventana
Nodos_de_Barras_EA,
%se consigue obtener la matriz BARRAS
```

```
for i=1:numero_barras
    contadornumerobarras=i;
    Nodos_de_Barras_EA

uiwait

global nodomenor nodomayor

BARRAS(i,:)=[nodomenor nodomayor];
barracol(i,:)=i;

end

%Se crea una matriz cuya primera columna indicara el numero de barras
de la
%estructura, y la matriz BARRAS, aportara el dato de los nodos de cada
%barra
tablabarras=[barracol BARRAS];

%se muestra la matriz tablabarras en la ventana Definicion EStructura
%Articulada
set(handles.barras_output,'data',tablabarras)

% --- Executes on button press in muelle_menu.
function muelle_menu_Callback(hObject, eventdata, handles)

global numero_barras Barramuelle numero_muelles contadorbarmueart
tablactek

%Mediante este bucle se obtiene la matriz Barramuelle
Barramuelle=zeros(numero_barras,1);
muelecol=[1:1:numero_barras];
muellecol=muellecol';

if numero_muelles==0

tablactek=[muellecol Barramuelle];

set(handles.valorctek_output,'data',tablactek)

elseif numero_muelles>0

    for i=1:numero_muelles
        contadorbarmueart=i;
        Valor_Cte_Rigidez_EA

uiwait

global valordek numerodemuelle
Barramuelle(numerodemuelle,:)=valordek;
muellecol(i,:)=i;
    end
end

%Se crea una matriz cuya primera columna indicara el numero de barras
de la
%estructura, y la matriz Barramuella que aportara el dato de la
constante
%de rigidez de cada barra
tablactek=[muellecol Barramuelle];
```

```
%Se muestra la matriz tablactek en la ventana Definicion Estructura
%Articulada
set(handles.valorctek_output,'data',tablactek)

% --- Executes on button press in coordnodos_menu.
function coordnodos_menu_Callback(hObject, eventdata, handles)

global numero_nodos Nodo numeronodocoordenadasart
tablacoordenadasnodos

%Se obtiene la matriz Nodo, con las coordenadas de cada nodo de la
%estructura
for i=1:numero_nodos
    numeronodocoordenadasart=i;
    Coordenadas_Nodos_EA

uiwait

global coordenadax coordenaday
nodocol(i,:)=i;
Nodo(i,:)=[coordenadax coordenaday];

end

%Se muestran los resultados en una tabla en pantalla, indicando que a
cada nodo el
%corresponde tales coordenadas
tablacoordenadasnodos=[nodocol Nodo];

set(handles.coordenadasnodos_output,'data',tablacoordenadasnodos)

% --- Executes on button press in restriccnodos_menu.
function restriccnodos_menu_Callback(hObject, eventdata, handles)

global numero_nodos RestriccionesNodo condicionesiniciales
numero_restricciones contadornumrestriccart tablarestricciones

%Se obtiene la matriz RestriccionesNodo, que nos indicara los grados
de
%libertad que tiene cada nodd de la estructura
RestriccionesNodo=ones(numero_nodos,2);
restriccolvector=[1:1:numero_nodos];
restriccol=restriccolvector';

for i=1:numero_restricciones
    contadornumrestriccart=i;
    Tipo_Apoyo_EA

uiwait

global desplazh desplazv numnodorestringido
RestriccionesNodo(numnodorestringido,:)=[desplazh desplazv];

end

%Se muestran los resultados obtenidos en una tabla en la ventana de
%Definicion Estructura Articulada
tablarestricciones=[restriccol RestriccionesNodo];
```

```
set(handles.desplazamientos_output,'data',tablarestricciones)

%Se obtiene el vector condicionesdecontorno, que ser necesario para
%obtener los desplazamientos mediante la instruccion ode45
contadorci=0;
for i=1:numero_nodos
    for I=1:2
        if RestriccionesNodo(i,I)==1
            contadorci=contadorci+1;
        end
    end
end
cond=contadorci*2;
condicionesiniciales=zeros(1,cond);

% --- Executes on button press in amortiguador_menu.
function amortiguador_menu_Callback(hObject, eventdata, handles)

global numero_barras Barraamortiguador numero_amortiguadores
contadorbaramoart tablactec

%Se obtiene el vector Barraamortiguador con los valores de la
constante de
%amortiguador de los amortiguadores que tenga la estructura
amortigco=[1:1:numero_barras];
amortiguadorcol=amortigco';
Barraamortiguador=zeros(numero_barras,1);

if numero_amortiguadores==0
tablactec=[amortiguadorcol Barraamortiguador];
set(handles.valorctec_output,'data',tablactec)

elseif numero_amortiguadores>0

    for i=1:numero_amortiguadores
        contadorbaramoart=i;
Valor_Cte_C_EA

uiwait

global valordec numerodeamortiguador
Barraamortiguador(numerodeamortiguador,:)=valordec;

end
end

%Se muestran los valores del vector Barraamortiguador en la ventana
%Definicion Estructura Articulada
tablactec=[amortiguadorcol Barraamortiguador];
set(handles.valorctec_output,'data',tablactec)

% --- Executes on button press in caract_menu.
function caract_menu_Callback(hObject, eventdata, handles)
% hObject    handle to caract_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
%Se abre la ventana Especificacion_Secciones
Especificacion_Secciones

uiwait

global E densidad A barracolespecificacionsecciones tablastecciones

%Se genera una matriz con los valores del modulo de elasticidad,
densidad
%area, y numero de barras
tablastecciones=[barracolespecificacionsecciones E densidad A];

%Se muestra la matriz tablastecciones en una tabla en la ventana
Definicion
%Estructura Articulada
set(handles.tablastecciones_output,'data',tablastecciones)

% --- Executes on button press in resolution_menu.
function resolution_menu_Callback(hObject, eventdata, handles)
% hObject    handle to resolution_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global t x
[t,x]= ode45 (@xprimaarticuladaconcoordenadasmuelleyamortiguador,
[0,(20*pi)/1000],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]);

global E A densidad omegaaa B BARRAS Nodo RestriccionesNodo
FuerzaNodos Barramuelle Barraamortiguador condicionesiniciales t x
tini tfin desfase
[t,x]= ode45
(@xprimaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ,
[tini,tfin],condicionesiniciales,[],E,A,densidad,omegaaa,B,BARRAS,Nodo
,RestriccionesNodo,FuerzaNodos,Barramuelle,Barraamortiguador,tfin,desf
ase);

function nodo_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodo_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodo_output as text
%        str2double(get(hObject,'String')) returns contents of
nodo_output as a double

% --- Executes during object creation, after setting all properties.
function nodo_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodo_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function nodos_output_Callback(hObject, eventdata, handles)
% hObject      handle to nodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodos_output as
text
%           str2double(get(hObject,'String')) returns contents of
nodos_output as a double

% --- Executes during object creation, after setting all properties.
function nodos_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function coordenadas_output_Callback(hObject, eventdata, handles)
% hObject      handle to coordenadas_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of coordenadas_output
as text
%           str2double(get(hObject,'String')) returns contents of
coordenadas_output as a double

% --- Executes during object creation, after setting all properties.
function coordenadas_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to coordenadas_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in numbarras_menu.
function numbarras_menu_Callback(hObject, eventdata, handles)
% hObject      handle to numbarras_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Barras_EA
Numero_Barras_EA
```

uiwait

```
global numero_barras
%Se muestra en la ventana Definicion Estructura Articulada el valor de
la
%variable numero_barras
set(handles.barra_output, 'String', numero_barras)
```

```
function barra_output_Callback(hObject, eventdata, handles)
% hObject    handle to barra_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of barra_output as
text
%          str2double(get(hObject, 'String')) returns contents of
barra_output as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function barra_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to barra_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function barras_output_Callback(hObject, eventdata, handles)
% hObject    handle to barras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject, 'String') returns contents of barras_output as
text
%          str2double(get(hObject, 'String')) returns contents of
barras_output as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function barras_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to barras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```



```
function uniones_output_Callback(hObject, eventdata, handles)
% hObject      handle to uniones_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of uniones_output as
text
%          str2double(get(hObject,'String')) returns contents of
uniones_output as a double

% --- Executes during object creation, after setting all properties.
function uniones_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to uniones_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodorestricc_output_Callback(hObject, eventdata, handles)
% hObject      handle to nodorestricc_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodorestricc_output
as text
%          str2double(get(hObject,'String')) returns contents of
nodorestricc_output as a double

% --- Executes during object creation, after setting all properties.
function nodorestricc_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodorestricc_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function restricc_output_Callback(hObject, eventdata, handles)
% hObject      handle to restricc_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of restricc_output as
text
%          str2double(get(hObject,'String')) returns contents of
```

restricc\_output as a double

```
% --- Executes during object creation, after setting all properties.
function restricc_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to restricc_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function nodofue_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodofue_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodofue_output as
text
%         str2double(get(hObject,'String')) returns contents of
nodofue_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function nodofue_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodofue_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function fue_output_Callback(hObject, eventdata, handles)
% hObject    handle to fue_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fue_output as text
%         str2double(get(hObject,'String')) returns contents of
fue_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function fue_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fue_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function barmue_output_Callback(hObject, eventdata, handles)
% hObject      handle to barmue_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of barmue_output as
text
%      str2double(get(hObject,'String')) returns contents of
barmue_output as a double

% --- Executes during object creation, after setting all properties.
function barmue_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to barmue_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valork_output_Callback(hObject, eventdata, handles)
% hObject      handle to valork_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valork_output as
text
%      str2double(get(hObject,'String')) returns contents of
valork_output as a double

% --- Executes during object creation, after setting all properties.
function valork_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valork_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function baramo_output_Callback(hObject, eventdata, handles)
% hObject      handle to baramo_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of baramo_output as
text
%           str2double(get(hObject,'String')) returns contents of
baramo_output as a double

% --- Executes during object creation, after setting all properties.
function baramo_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to baramo_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorc_output_Callback(hObject, eventdata, handles)
% hObject      handle to valorc_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorc_output as
text
%           str2double(get(hObject,'String')) returns contents of
valorc_output as a double

% --- Executes during object creation, after setting all properties.
function valorc_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valorc_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valordiametro_output_Callback(hObject, eventdata, handles)
% hObject      handle to valordiametro_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valordiametro_output as text
%           str2double(get(hObject,'String')) returns contents of
valordiametro_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function valordiametro_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valordiametro_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valordensidad_output_Callback(hObject, eventdata, handles)
% hObject    handle to valordensidad_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valordensidad_output as text
%         str2double(get(hObject,'String')) returns contents of
valordensidad_output as a double

% --- Executes during object creation, after setting all properties.
function valordensidad_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valordensidad_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorE_output_Callback(hObject, eventdata, handles)
% hObject    handle to valorE_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorE_output as
text
%         str2double(get(hObject,'String')) returns contents of
valorE_output as a double

% --- Executes during object creation, after setting all properties.
function valorE_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valorE_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valoromega_output_Callback(hObject, eventdata, handles)
% hObject      handle to valoromega_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valoromega_output
as text
%      str2double(get(hObject,'String')) returns contents of
valoromega_output as a double

% --- Executes during object creation, after setting all properties.
function valoromega_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valoromega_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valordesfase_output_Callback(hObject, eventdata, handles)
% hObject      handle to valordesfase_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valordesfase_output
as text
%      str2double(get(hObject,'String')) returns contents of
valordesfase_output as a double

% --- Executes during object creation, after setting all properties.
function valordesfase_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valordesfase_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in numrestricc_menu.
```

```
function numrestricc_menu_Callback(hObject, eventdata, handles)
% hObject      handle to numrestricc_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Nodos_Restringidos_EA
Numero_Nodos_Restringidos_EA

uiwait

global numero_restricciones
%Se muestra en la ventana Definicion Estructura Articulada el valor
de la
%variable numero_restricciones
set(handles.restricci_output,'String',numero_restricciones)

function restricci_output_Callback(hObject, eventdata, handles)
% hObject      handle to restricci_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of restricci_output as
text
%          str2double(get(hObject,'String')) returns contents of
restricci_output as a double

% --- Executes during object creation, after setting all properties.
function restricci_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to restricci_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Al pulsar el boton cerrar, se cierra la ventana
%Definicion_Estructura_Articulada

close Definicion_Estructura_Articulada

% --- Executes during object creation, after setting all properties.
function numfuerzas_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numfuerzas_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in nummuelles_menu.
function nummuelles_menu_Callback(hObject, eventdata, handles)
% hObject     handle to nummuelles_menu (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Muelles_EA
Numero_Muelles_EA

uiwait

global numero_muelles
%Se muestra en la ventana Definicion Estructura Articulada el valor de
la
%variable numero_muelles
set(handles.nummuelles_output,'String',numero_muelles)

function nummuelles_output_Callback(hObject, eventdata, handles)
% hObject     handle to nummuelles_output (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nummuelles_output
as text
%     str2double(get(hObject,'String')) returns contents of
nummuelles_output as a double

% --- Executes during object creation, after setting all properties.
function nummuelles_output_CreateFcn(hObject, eventdata, handles)
% hObject     handle to nummuelles_output (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in numamortiguadores_menu.
function numamortiguadores_menu_Callback(hObject, eventdata, handles)
% hObject     handle to numamortiguadores_menu (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Amortiguadores_EA
Numero_Amortiguadores_EA
```



uiwait

```
global numero_amortiguadores
```

```
%Se muestra en la ventana Definicion Estructura Articulada el valor de la
```

```
%variable numero_amortiguadores
```

```
set(handles.numamortiguadores_output, 'String', numero_amortiguadores)
```

```
function numamortiguadores_output_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to numamortiguadores_output (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject, 'String') returns contents of numamortiguadores_output as text
```

```
%          str2double(get(hObject, 'String')) returns contents of numamortiguadores_output as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function numamortiguadores_output_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to numamortiguadores_output (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
```

```
    set(hObject, 'BackgroundColor', 'white');
```

```
end
```

```
function valorI_output_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to valorI_output (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject, 'String') returns contents of valorI_output as text
```

```
%          str2double(get(hObject, 'String')) returns contents of valorI_output as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function valorI_output_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to valorI_output (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorasignacionbarra_output_Callback(hObject, eventdata,
handles)
% hObject      handle to valorasignacionbarra_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valorasignacionbarra_output as text
%          str2double(get(hObject,'String')) returns contents of
valorasignacionbarra_output as a double

% --- Executes during object creation, after setting all properties.
function valorasignacionbarra_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to valorasignacionbarra_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

## Definición estructura reticulada

```
function varargout = Definicion_Estructura_Reticulada(varargin)
% DEFINICION_ESTRUCTURA_RETICULADA M-file for
Definicion_Estructura_Reticulada.fig
%     DEFINICION_ESTRUCTURA_RETICULADA, by itself, creates a new
DEFINICION_ESTRUCTURA_RETICULADA or raises the existing
%     singleton*.
%
%     H = DEFINICION_ESTRUCTURA_RETICULADA returns the handle to a
new DEFINICION_ESTRUCTURA_RETICULADA or the handle to
%     the existing singleton*.
%
%
DEFINICION_ESTRUCTURA_RETICULADA('CALLBACK',hObject,eventData,handles,
...) calls the local
%     function named CALLBACK in DEFINICION_ESTRUCTURA_RETICULADA.M
with the given input arguments.
%
%     DEFINICION_ESTRUCTURA_RETICULADA('Property','Value',...)
creates a new DEFINICION_ESTRUCTURA_RETICULADA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Definicion_Estructura_Reticulada_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Definicion_Estructura_Reticulada_OpeningFcn via varargin.
```

```
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Definicion_Estructura_Reticulada

% Last Modified by GUIDE v2.5 18-Nov-2011 18:28:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Definicion_Estructura_Reticulada_OpeningFcn, ...
                  'gui_OutputFcn',    @Definicion_Estructura_Reticulada_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Definicion_Estructura_Reticulada is made
visible.
function Definicion_Estructura_Reticulada_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to
Definicion_Estructura_Reticulada (see VARARGIN)

% Choose default command line output for
Definicion_Estructura_Reticulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Definicion_Estructura_Reticulada wait for user response
(see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout =
Definicion_Estructura_Reticulada_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_nodos tablabarras tablamuelles tablacoordenadas
tablarestricciones tablaamortiguadores tablasecciones numero_barras
numero_restricciones numero_muelles numero_amortiguadores

%Se muestra en la ventana Definicion Estructura Reticulada el valor de
las
%variables globales
set(handles.nodor_output,'String',numero_nodos)
set(handles.tablabarras_output,'data',tablabarras)
set(handles.valorctek_output,'data',tablamuelles)
set(handles.coordenadas_output,'data',tablacoordenadas)
set(handles.tablarestricc_output,'data',tablarestricciones)
set(handles.valorctec_output,'data',tablaamortiguadores)
set(handles.secciones_output,'data',tablasecciones)
set(handles.barrar_output,'String',numero_barras)
set(handles.restriccir_output,'String',numero_restricciones)
set(handles.nummuellesr_output,'String',numero_muelles)
set(handles.numamortiguadoresr_output,'String',numero_amortiguadores)

% --- Executes on button press in numnodosr_menu.
function numnodosr_menu_Callback(hObject, eventdata, handles)
% hObject handle to numnodosr_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Nodos_ER
Numero_Nodos_ER

uiwait

global numero_nodos
%Se muestra el valor de la variable numero_nodos en la ventana
Definicion
%Estructura Reticulada
set(handles.nodor_output,'String',numero_nodos)

% --- Executes on button press in barrasr_menu.
function barrasr_menu_Callback(hObject, eventdata, handles)

global numero_barras BARRAS contadornumarret tablabarras

%Se abre la ventana Nodos_de_Barras_ER tantas veces como numero de
barras
%tenga la estructura obteniendose el valor de la matriz BARRAS
for i=1:numero_barras
    contadornumarret=i;
    Nodos_de_Barras_ER

uiwait

global nodomenor nodomayor
BARRAS(i,:)=[nodomenor nodomayor];
barracol(i,:)=i;
```

```
end
tablabarras=[barracol BARRAS];
%Se muestra en la ventana Definicion Estructura Reticulada el valor de
la
%matriz BARRAS
set(handles.tablabarras_output,'data',tablabarras)

% --- Executes on button press in mueller_menu.
function mueller_menu_Callback(hObject, eventdata, handles)

global numero_barras Barramuelle numero_muelles contadormueret
tablamuelles

%Se obtiene el vector Barramuelle
Barramuelle=zeros(numero_barras,1);

muelecol=[1:1:numero_barras];
muellecol=muelecol';
if numero_muelles==0

tablamuelles=[muellecol Barramuelle];
set(handles.valorctek_output,'data',tablamuelles)

elseif numero_muelles>0

    for i=1:numero_muelles
        contadormueret=i;
        Valor_Cte_Rigidez_ER

        uiwait

        global valordek numerodemuelle
        Barramuelle(numerodemuelle,:)=valordek;
        muellecol(i,:)=i;

    end

end

tablamuelles=[muellecol Barramuelle];
%Se muestra en pantalla los valores del vector Barramuelle
set(handles.valorctek_output,'data',tablamuelles)

% --- Executes on button press in coordnodosr_menu.
function coordnodosr_menu_Callback(hObject, eventdata, handles)

global numero_nodos Nodo contadorcoordnodret tablacoordenadas

%Se obtiene la matriz Nodo a partir de los valores que se introducirán
en
%la ventana Coordenadas_Nodos_ER
for i=1:numero_nodos
    contadorcoordnodret=i;
    Coordenadas_Nodos_ER

    uiwait

    global coordenadax coordenaday
    Nodo(i,:)=[coordenadax coordenaday];
    nodocol(i,:)=i;
```

```
end
tablacoordenadas=[nodocol Nodo];
%Se muestran en la ventana Definicion Estructura Reticulada los
valores de
%la matriz Nodo
set(handles.coordenadas_output,'data',tablacoordenadas)

% --- Executes on button press in restriccnodosr_menu.
function restriccnodosr_menu_Callback(hObject, eventdata, handles)

global numero_nodos RestriccionesNodo numero_restricciones
condicionesiniciales contadorrestret tablarestricciones

%se obtiene la matriz RestriccionesNodo a partir de los datos
introducidos
%en la ventana Tipo_Apoyo_ER
RestriccionesNodo=ones(numero_nodos,3);
restriccolvector=[1:1:numero_nodos];
restriccol=restriccolvector';

for i=1:numero_restricciones
    contadorrestret=i;
    Tipo_Apoyo_ER

uiwait

global desplazh desplazv giro numnodorestringido
RestriccionesNodo(numnodorestringido,:)=[desplazh desplazv giro];

end
tablarestricciones=[restriccol RestriccionesNodo];
%Se muestran los valores de la matriz RestriccionesNodo en la ventana
%Definicion Estructura Reticulada
set(handles.tablarestricc_output,'data',tablarestricciones)

%Se obtienen las condiciones iniciales que servira como dato a la
%instruccion ode45 a la hora de obtener los desplazamientos
contadorci=0;
for i=1:numero_nodos
    for I=1:3
        if RestriccionesNodo(i,I)==1
            contadorci=contadorci+1;
        end
    end
end
cond=contadorci*2;
condicionesiniciales=zeros(1,cond);

% --- Executes on button press in amortiguadorr_menu.
function amortiguadorr_menu_Callback(hObject, eventdata, handles)
% hObject handle to amortiguadorr_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global numero_barras Barraamortiguador numero_amortiguadores
contadoramorret tablaamortiguadores

%Se obtiene el vector Barraamortiguador a partir de los valores
```

```
%introducidos en la ventana Valor_Cte_C_ER
amortigco=[1:1:numero_barras];
amortiguadorcol=amortigco';
Barraamortiguador=zeros(numero_barras,1);
if numero_amortiguadores==0

    tablaamortiguadores=[amortiguadorcol Barraamortiguador];

    set(handles.valorctec_output,'data',tablaamortiguadores)

elseif numero_amortiguadores>0
    for i=1:numero_amortiguadores
        contadoramorret=i;
Valor_Cte_C_ER

uiwait

global valordec numerodeamortiguador
Barraamortiguador(numerodeamortiguador,:)=valordec;
amortiguadorcol(i,:)=i;

    end
end
tablaamortiguadores=[amortiguadorcol Barraamortiguador];
    %Se muestra en la ventana Definicion Estructura Reticulada los
valores
    %del vector Barraamortiguaodor
    set(handles.valorctec_output,'data',tablaamortiguadores)

% --- Executes on button press in caractr_menu.
function caractr_menu_Callback(hObject, eventdata, handles)
% hObject      handle to caractr_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana Especificacion_Secciones_ER
Especificacion_Secciones_ER

uiwait

global E densidad A I barracolespecificacionmateriales tablasecciones

tablasecciones=[barracolespecificacionmateriales E A I densidad];
%Se muestra en pantalla los valores de la matriz tablasecciones que
%contendra los valores del modulo de elasticidad, area, momento de
%ineracia, y densidad de cada barra.
set(handles.secciones_output,'data',tablasecciones)

function nodor_output_Callback(hObject, eventdata, handles)
% hObject      handle to nodor_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodor_output as
text
%          str2double(get(hObject,'String')) returns contents of
nodor_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function nodor_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodor_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodosr_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodosr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodosr_output as
text
%         str2double(get(hObject,'String')) returns contents of
nodosr_output as a double

% --- Executes during object creation, after setting all properties.
function nodosr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodosr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function coordenadasr_output_Callback(hObject, eventdata, handles)
% hObject    handle to coordenadasr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of coordenadasr_output
as text
%         str2double(get(hObject,'String')) returns contents of
coordenadasr_output as a double

% --- Executes during object creation, after setting all properties.
function coordenadasr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to coordenadasr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```



```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
% --- Executes on button press in numbarrasr_menu.  
function numbarrasr_menu_Callback(hObject, eventdata, handles)  
% hObject    handle to numbarrasr_menu (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
%Se abre la ventana Numero_Barras_ER  
Numero_Barras_ER  
  
uiwait  
  
global numero_barras  
%Se muestra en la ventana Definicion Estructura Reticulada el valor de  
la  
%variable numero_barras  
set(handles.barrar_output,'String',numero_barras)  
  
function barrar_output_Callback(hObject, eventdata, handles)  
% hObject    handle to barrar_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of barrar_output as  
text  
%         str2double(get(hObject,'String')) returns contents of  
barrar_output as a double  
  
% --- Executes during object creation, after setting all properties.  
function barrar_output_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to barrar_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
function barrasr_output_Callback(hObject, eventdata, handles)  
% hObject    handle to barrasr_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of barrasr_output as  
text  
%         str2double(get(hObject,'String')) returns contents of  
barrasr_output as a double  
  
% --- Executes during object creation, after setting all properties.
```

```
function barrasr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to barrasr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function unionesr_output_Callback(hObject, eventdata, handles)
% hObject    handle to unionesr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of unionesr_output as
text
%       str2double(get(hObject,'String')) returns contents of
unionesr_output as a double

% --- Executes during object creation, after setting all properties.
function unionesr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to unionesr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodorestriccr_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodorestriccr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nodorestriccr_output as text
%       str2double(get(hObject,'String')) returns contents of
nodorestriccr_output as a double

% --- Executes during object creation, after setting all properties.
function nodorestriccr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodorestriccr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function restriccr_output_Callback(hObject, eventdata, handles)
% hObject      handle to restriccr_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of restriccr_output as
text
%           str2double(get(hObject,'String')) returns contents of
restriccr_output as a double

% --- Executes during object creation, after setting all properties.
function restriccr_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to restriccr_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodofuer_output_Callback(hObject, eventdata, handles)
% hObject      handle to nodofuer_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodofuer_output as
text
%           str2double(get(hObject,'String')) returns contents of
nodofuer_output as a double

% --- Executes during object creation, after setting all properties.
function nodofuer_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodofuer_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function barmuer_output_Callback(hObject, eventdata, handles)
% hObject      handle to barmuer_output (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of barmuer_output as
text
%          str2double(get(hObject,'String')) returns contents of
barmuer_output as a double

% --- Executes during object creation, after setting all properties.
function barmuer_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to barmuer_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorKr_output_Callback(hObject, eventdata, handles)
% hObject      handle to valorKr_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorKr_output as
text
%          str2double(get(hObject,'String')) returns contents of
valorKr_output as a double

% --- Executes during object creation, after setting all properties.
function valorKr_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valorKr_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function baramor_output_Callback(hObject, eventdata, handles)
% hObject      handle to baramor_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of baramor_output as
text
%          str2double(get(hObject,'String')) returns contents of
baramor_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function baramor_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to baramor_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function valorcr_output_Callback(hObject, eventdata, handles)
% hObject    handle to valorcr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorcr_output as
text
%       str2double(get(hObject,'String')) returns contents of
valorcr_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function valorcr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valorcr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function valordiametror_output_Callback(hObject, eventdata, handles)
% hObject    handle to valordiametror_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valordiametror_output as text
%       str2double(get(hObject,'String')) returns contents of
valordiametror_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function valordiametror_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valordiametror_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
function valordensidadr_output_Callback(hObject, eventdata, handles)  
% hObject    handle to valordensidadr_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of  
valordensidadr_output as text  
%         str2double(get(hObject,'String')) returns contents of  
valordensidadr_output as a double  
  
% --- Executes during object creation, after setting all properties.  
function valordensidadr_output_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to valordensidadr_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
function valorEr_output_Callback(hObject, eventdata, handles)  
% hObject    handle to valorEr_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of valorEr_output as  
text  
%         str2double(get(hObject,'String')) returns contents of  
valorEr_output as a double  
  
% --- Executes during object creation, after setting all properties.  
function valorEr_output_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to valorEr_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%         See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
function valoromegar_output_Callback(hObject, eventdata, handles)  
% hObject    handle to valoromegar_output (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valoromegar_output
as text
%          str2double(get(hObject,'String')) returns contents of
valoromegar_output as a double

% --- Executes during object creation, after setting all properties.
function valoromegar_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valoromegar_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valordesfaser_output_Callback(hObject, eventdata, handles)
% hObject      handle to valordesfaser_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valordesfaser_output as text
%          str2double(get(hObject,'String')) returns contents of
valordesfaser_output as a double

% --- Executes during object creation, after setting all properties.
function valordesfaser_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to valordesfaser_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in numrestriccr_menu.
function numrestriccr_menu_Callback(hObject, eventdata, handles)
% hObject      handle to numrestriccr_menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana Numero_Nodos_Restringidos_ER
Numero_Nodos_Restringidos_ER

uiwait
```

```
global numero_restricciones

%Se muestra en pantalla el valor de la variable numero_restricciones
set(handles.restriccir_output, 'String', numero_restricciones)

function restriccir_output_Callback(hObject, eventdata, handles)
% hObject    handle to restriccir_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of restriccir_output
as text
%          str2double(get(hObject, 'String')) returns contents of
restriccir_output as a double

% --- Executes during object creation, after setting all properties.
function restriccir_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to restriccir_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
%Se cierra la ventana Definicion_Estructura_Reticulada

close Definicion_Estructura_Reticulada

% --- Executes on button press in nummuellesr_menu.
function nummuellesr_menu_Callback(hObject, eventdata, handles)
%Se abre la ventana Numero_Muelles_ER

Numero_Muelles_ER

uiwait

global numero_muelles
%Se muestra en pantalla el valor de la variable numero_muelles
set(handles.nummuellesr_output, 'String', numero_muelles)

function nummuellesr_output_Callback(hObject, eventdata, handles)
% hObject    handle to nummuellesr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of nummuellesr_output
as text
%          str2double(get(hObject, 'String')) returns contents of
```



nummuellesr\_output as a double

```
% --- Executes during object creation, after setting all properties.
function nummuellesr_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nummuellesr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in numamortiguadoresr_menu.
function numamortiguadoresr_menu_Callback(hObject, eventdata, handles)
%Se abre la ventana Numero_Amortiguadores_ER

Numero_Amortiguadores_ER

uiwait

global numero_amortiguadores
%Se mostrara en la ventana Definicoiin Estructura Reticulada el valor
de la
%variable numero_amortiguadores
set(handles.numamortiguadoresr_output,'String',numero_amortiguadores)

function numamortiguadoresr_output_Callback(hObject, eventdata,
handles)
% hObject    handle to numamortiguadoresr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numamortiguadoresr_output as text
%       str2double(get(hObject,'String')) returns contents of
numamortiguadoresr_output as a double

% --- Executes during object creation, after setting all properties.
function numamortiguadoresr_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to numamortiguadoresr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorir_output_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to valorir_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorir_output as
text
%          str2double(get(hObject,'String')) returns contents of
valorir_output as a double

% --- Executes during object creation, after setting all properties.
function valorir_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valorir_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valorasignacionbarra_output_Callback(hObject, eventdata,
handles)
% hObject    handle to valorasignacionbarra_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
valorasignacionbarra_output as text
%          str2double(get(hObject,'String')) returns contents of
valorasignacionbarra_output as a double

% --- Executes during object creation, after setting all properties.
function valorasignacionbarra_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to valorasignacionbarra_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Introducir el número de nodos de la estructura reticulada

```
function varargout = Numero_Nodos_ER(varargin)
% NUMERO_NODOS_ER M-file for Numero_Nodos_ER.fig
%     NUMERO_NODOS_ER, by itself, creates a new NUMERO_NODOS_ER or
raises the existing
%     singleton*.
%
%     H = NUMERO_NODOS_ER returns the handle to a new NUMERO_NODOS_ER
```

```
or the handle to
%     the existing singleton*.
%
%     NUMERO_NODOS_ER('CALLBACK', hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in NUMERO_NODOS_ER.M with the given
input arguments.
%
%     NUMERO_NODOS_ER('Property','Value',...) creates a new
NUMERO_NODOS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Nodos_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_Nodos_ER_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Nodos_ER

% Last Modified by GUIDE v2.5 19-Nov-2011 17:23:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Nodos_ER_OpeningFcn, ...
                  'gui_OutputFcn',  @Numero_Nodos_ER_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Nodos_ER is made visible.
function Numero_Nodos_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Nodos_ER (see VARARGIN)

% Choose default command line output for Numero_Nodos_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes Numero_Nodos_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Nodos_ER_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrarr.
function cerrarr_Callback(hObject, eventdata, handles)
%Se cierra la ventana Numero_Nodos_ER
close Numero_Nodos_ER

function numnodosr_input_Callback(hObject, eventdata, handles)
% hObject      handle to numnodosr_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodosr_input as
text
%           str2double(get(hObject,'String')) returns contents of
numnodosr_input as a double

% --- Executes during object creation, after setting all properties.
function numnodosr_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numnodosr_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numnodosr_output_Callback(hObject, eventdata, handles)
% hObject      handle to numnodosr_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodosr_output as
text
%           str2double(get(hObject,'String')) returns contents of
numnodosr_output as a double

% --- Executes during object creation, after setting all properties.
function numnodosr_output_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to numnodosr_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptarr.
function aceptarr_Callback(hObject, eventdata, handles)

%Se hace global la variable numero_nodos
global numero_nodos
%Se le asigna el valor introducido en la edit text de tag
numnodosr_input a
%la vairble numero_nodos
numero_nodos=str2double(get(handles.numnodosr_input,'String'));
%Se muestra en la ventana Numero_Nodos_ER el valor de la variable
%numero_nodos
set(handles.numnodosr_output,'String',numero_nodos);
```

### Introducir el número de nodos de la estructura articulada

```
function varargout = Numero_Nodos_EA(varargin)
% NUMERO_NODOS_EA M-file for Numero_Nodos_EA.fig
%     NUMERO_NODOS_EA, by itself, creates a new NUMERO_NODOS_EA or
raises the existing
%     singleton*.
%
%     H = NUMERO_NODOS_EA returns the handle to a new NUMERO_NODOS_EA
or the handle to
%     the existing singleton*.
%
%     NUMERO_NODOS_EA('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in NUMERO_NODOS_EA.M with the given
input arguments.
%
%     NUMERO_NODOS_EA('Property','Value',...) creates a new
NUMERO_NODOS_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Nodos_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_Nodos_EA_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Nodos_EA
```

```
% Last Modified by GUIDE v2.5 18-Nov-2011 18:39:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Nodos_EA_OpeningFcn, ...
                  'gui_OutputFcn',  @Numero_Nodos_EA_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Numero_Nodos_EA is made visible.
function Numero_Nodos_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Numero_Nodos_EA (see VARARGIN)

% Choose default command line output for Numero_Nodos_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Nodos_EA wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Numero_Nodos_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Se cierra la ventana
close Numero_Nodos_EA
```

```
function numnodos_input_Callback(hObject, eventdata, handles)
% hObject      handle to numnodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodos_input as
text
%           str2double(get(hObject,'String')) returns contents of
numnodos_input as a double

% --- Executes during object creation, after setting all properties.
function numnodos_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numnodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numnodos_output_Callback(hObject, eventdata, handles)
% hObject      handle to numnodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodos_output as
text
%           str2double(get(hObject,'String')) returns contents of
numnodos_output as a double

% --- Executes during object creation, after setting all properties.
function numnodos_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numnodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar_numnodos.
function aceptar_numnodos_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar_numnodos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se hace global la variable numero_nodos
global numero_nodos

%Se le asigna el valor introducido en la edit text de tag
```

```
numnodosr_input a
%la vairble numero_nodos
numero_nodos=str2double(get(handles.numnodos_input,'String'));

%Se muestra en la ventana Numero_Nodos_ER el valor de la variable
%numero_nodos
set(handles.numnodos_output,'String',numero_nodos);
```

### Definición de los nodos de cada barra de la estructura reticulada

```
function varargout = Nodos_de_Barras_ER(varargin)
% NODOS_DE_BARRAS_ER M-file for Nodos_de_Barras_ER.fig
%     NODOS_DE_BARRAS_ER, by itself, creates a new NODOS_DE_BARRAS_ER
or raises the existing
%     singleton*.
%
%
%     H = NODOS_DE_BARRAS_ER returns the handle to a new
NODOS_DE_BARRAS_ER or the handle to
%     the existing singleton*.
%
%     NODOS_DE_BARRAS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NODOS_DE_BARRAS_ER.M with the given
input arguments.
%
%     NODOS_DE_BARRAS_ER('Property','Value',...) creates a new
NODOS_DE_BARRAS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Nodos_de_Barras_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Nodos_de_Barras_ER_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Nodos_de_Barras_ER

% Last Modified by GUIDE v2.5 20-Nov-2011 14:47:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Nodos_de_Barras_ER_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Nodos_de_Barras_ER_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```



```
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Nodos_de_Barras_ER is made visible.
function Nodos_de_Barras_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Nodos_de_Barras_ER (see
VARARGIN)

% Choose default command line output for Nodos_de_Barras_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Nodos_de_Barras_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Nodos_de_Barras_ER_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadornumbarret

set(handles.contadornumbarret_output, 'String', contadornumbarret)

function nodomen_input_Callback(hObject, eventdata, handles)
% hObject    handle to nodomen_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of nodomen_input as
text
%         str2double(get(hObject, 'String')) returns contents of
nodomen_input as a double

% --- Executes during object creation, after setting all properties.
function nodomen_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodomen_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
function nodomay_input_Callback(hObject, eventdata, handles)  
% hObject      handle to nodomay_input (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of nodomay_input as  
text  
%           str2double(get(hObject,'String')) returns contents of  
nodomay_input as a double  
  
% --- Executes during object creation, after setting all properties.  
function nodomay_input_CreateFcn(hObject, eventdata, handles)  
% hObject      handle to nodomay_input (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%           See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
% --- Executes on button press in aceptar.  
function aceptar_Callback(hObject, eventdata, handles)  
% hObject      handle to aceptar (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
global nodomenor nodomayor  
  
%Se asigna el valor introducido en cada uno de los edit text a las  
%variables nodomenor y nodomayor  
nodomenor=str2double(get(handles.nodomen_input,'String'));  
nodomayor=str2double(get(handles.nodomay_input,'String'));  
  
%Se cierra la ventana  
close Nodos_de_Barras_ER  
  
function contadornumbarret_output_Callback(hObject, eventdata,  
handles)  
% hObject      handle to contadornumbarret_output (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of  
contadornumbarret_output as text  
%           str2double(get(hObject,'String')) returns contents of  
contadornumbarret_output as a double  
  
% --- Executes during object creation, after setting all properties.
```

```
function contadornumbarret_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to contadornumbarret_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Definición de los nodos de cada barra de la estructura articulada

```
function varargout = Nodos_de_Barras_EA(varargin)
% NODOS_DE_BARRAS_EA M-file for Nodos_de_Barras_EA.fig
%     NODOS_DE_BARRAS_EA, by itself, creates a new NODOS_DE_BARRAS_EA
or raises the existing
%     singleton*.
%
%     H = NODOS_DE_BARRAS_EA returns the handle to a new
NODOS_DE_BARRAS_EA or the handle to
%     the existing singleton*.
%
%     NODOS_DE_BARRAS_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NODOS_DE_BARRAS_EA.M with the given
input arguments.
%
%     NODOS_DE_BARRAS_EA('Property','Value',...) creates a new
NODOS_DE_BARRAS_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Nodos_de_Barras_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Nodos_de_Barras_EA_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Nodos_de_Barras_EA

% Last Modified by GUIDE v2.5 20-Nov-2011 12:54:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Nodos_de_Barras_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Nodos_de_Barras_EA_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
```

```
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Nodos_de_Barras_EA is made visible.
function Nodos_de_Barras_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Nodos_de_Barras_EA (see
VARARGIN)

% Choose default command line output for Nodos_de_Barras_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Nodos_de_Barras_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Nodos_de_Barras_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadornumerobarras

set(handles.numerobarras_output, 'String', contadornumerobarras)

function nodomen_input_Callback(hObject, eventdata, handles)
% hObject    handle to nodomen_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of nodomen_input as
text
%         str2double(get(hObject, 'String')) returns contents of
nodomen_input as a double

% --- Executes during object creation, after setting all properties.
function nodomen_input_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to nodomen_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodomay_input_Callback(hObject, eventdata, handles)
% hObject    handle to nodomay_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodomay_input as
text
%         str2double(get(hObject,'String')) returns contents of
nodomay_input as a double

% --- Executes during object creation, after setting all properties.
function nodomay_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodomay_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global nodomenor nodomayor

%Se asigna el valor introducido en cada uno de los edit text a las
%variables nodomenor y nodomayor
nodomenor=str2double(get(handles.nodomen_input,'String'));
nodomayor=str2double(get(handles.nodomay_input,'String'));
%Se cierra la ventana
close Nodos_de_Barras_EA

function numerobarras_output_Callback(hObject, eventdata, handles)
% hObject    handle to numerobarras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numerobarras_output
as text
```

```
%      str2double(get(hObject,'String')) returns contents of
numerobarras_output as a double

% --- Executes during object creation, after setting all properties.
function numerobarras_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numerobarras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Valor de la constante de rigidez del muelle de la estructura reticulada

```
function varargout = Valor_Cte_Rigidez_ER(varargin)
% VALOR_CTE_RIGIDEZ_ER M-file for Valor_Cte_Rigidez_ER.fig
%     VALOR_CTE_RIGIDEZ_ER, by itself, creates a new
VALOR_CTE_RIGIDEZ_ER or raises the existing
%     singleton*.
%
%     H = VALOR_CTE_RIGIDEZ_ER returns the handle to a new
VALOR_CTE_RIGIDEZ_ER or the handle to
%     the existing singleton*.
%
%     VALOR_CTE_RIGIDEZ_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in VALOR_CTE_RIGIDEZ_ER.M with the
given input arguments.
%
%     VALOR_CTE_RIGIDEZ_ER('Property','Value',...) creates a new
VALOR_CTE_RIGIDEZ_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Valor_Cte_Rigidez_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Valor_Cte_Rigidez_ER_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Valor_Cte_Rigidez_ER

% Last Modified by GUIDE v2.5 21-Nov-2011 18:29:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Valor_Cte_Rigidez_ER_OpeningFcn,
```

```
...
                                'gui_OutputFcn', @Valor_Cte_Rigidez_ER_OutputFcn,
...
                                'gui_LayoutFcn', [] , ...
                                'gui_Callback', []];
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Valor_Cte_Rigidez_ER is made visible.
function Valor_Cte_Rigidez_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valor_Cte_Rigidez_ER (see
VARARGIN)

% Choose default command line output for Valor_Cte_Rigidez_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Valor_Cte_Rigidez_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Valor_Cte_Rigidez_ER_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadormueret
%Se muestra en pantalla el valor de la variable contadormueret
set(handles.contadormueret_output, 'String', contadormueret)

function ctek_input_Callback(hObject, eventdata, handles)
% hObject      handle to ctek_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of ctek_input as text
%        str2double(get(hObject, 'String')) returns contents of
%        ctek_input as a double
```

```
% --- Executes during object creation, after setting all properties.
function ctek_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ctek_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global valordek numerodemuelle

%Se asigna el valor introducido en los edit text de la ventana a las
%variables numerodemuelles y valordek
numerodemuelle=str2double(get(handles.numerodemuelle_input,'String'));
valordek=str2double(get(handles.cktek_input,'String'));
%Se cierra la ventana
close Valor_Cte_Rigidez_ER

function numerodemuelle_input_Callback(hObject, eventdata, handles)
% hObject    handle to numerodemuelle_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numerodemuelle_input as text
%         str2double(get(hObject,'String')) returns contents of
numerodemuelle_input as a double

% --- Executes during object creation, after setting all properties.
function numerodemuelle_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numerodemuelle_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadormueret_output_Callback(hObject, eventdata, handles)
% hObject    handle to contadormueret_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
```



```
contadormueret_output as text
%      str2double(get(hObject,'String')) returns contents of
contadormueret_output as a double

% --- Executes during object creation, after setting all properties.
function contadormueret_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to contadormueret_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Valor_Cte_Rigidez_ER
```

### Valor de la constante de rigidez de la estructura articulada

```
function varargout = Valor_Cte_Rigidez_EA(varargin)
% VALOR_CTE_RIGIDEZ_EA M-file for Valor_Cte_Rigidez_EA.fig
%      VALOR_CTE_RIGIDEZ_EA, by itself, creates a new
VALOR_CTE_RIGIDEZ_EA or raises the existing
%      singleton*.
%
%      H = VALOR_CTE_RIGIDEZ_EA returns the handle to a new
VALOR_CTE_RIGIDEZ_EA or the handle to
%      the existing singleton*.
%
%      VALOR_CTE_RIGIDEZ_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in VALOR_CTE_RIGIDEZ_EA.M with the
given input arguments.
%
%      VALOR_CTE_RIGIDEZ_EA('Property','Value',...) creates a new
VALOR_CTE_RIGIDEZ_EA or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Valor_Cte_Rigidez_EA_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to Valor_Cte_Rigidez_EA_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help
Valor_Cte_Rigidez_EA

% Last Modified by GUIDE v2.5 21-Nov-2011 18:28:43

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Valor_Cte_Rigidez_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',    @Valor_Cte_Rigidez_EA_OutputFcn,
                  ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Valor_Cte_Rigidez_EA is made visible.
function Valor_Cte_Rigidez_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valor_Cte_Rigidez_EA (see
VARARGIN)

% Choose default command line output for Valor_Cte_Rigidez_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Valor_Cte_Rigidez_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Valor_Cte_Rigidez_EA_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorbarmueart

%Se muestra en pantalla el valor de la variable contadorbarmueart
set(handles.barracontadorbarmueart_output, 'String', contadorbarmueart)
```

```
function ctek_input_Callback(hObject, eventdata, handles)
% hObject      handle to ctek_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ctek_input as text
%          str2double(get(hObject,'String')) returns contents of
%          ctek_input as a double

% --- Executes during object creation, after setting all properties.
function ctek_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ctek_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se convierten en variables globales
global valordek numerodemuelle

%Se asignan los valores introducidos en los edit text de la ventana a
las
%variables numerodemuelle y valordek
numerodemuelle=str2double(get(handles.numerodemuelle_input,'String'));
valordek=str2double(get(handles.cktek_input,'String'));
%Se cierra la ventana Valor_Cte_Rigidez_EA
close Valor_Cte_Rigidez_EA

function numerodemuelle_input_Callback(hObject, eventdata, handles)
% hObject      handle to numerodemuelle_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numerodemuelle_input as text
%          str2double(get(hObject,'String')) returns contents of
numerodemuelle_input as a double

% --- Executes during object creation, after setting all properties.
function numerodemuelle_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numerodemuelle_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function barracontadorbarmueart_output_Callback(hObject, eventdata,
handles)
% hObject      handle to barracontadorbarmueart_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
barracontadorbarmueart_output as text
%      str2double(get(hObject,'String')) returns contents of
barracontadorbarmueart_output as a double

% --- Executes during object creation, after setting all properties.
function barracontadorbarmueart_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to barracontadorbarmueart_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana
close Valor_Cte_Rigidez_EA
```

### Coordenadas de cada nodo de la estructura reticulada

```
function varargout = Coordenadas_Nodos_ER(varargin)
% COORDENADAS_NODOS_ER M-file for Coordenadas_Nodos_ER.fig
%      COORDENADAS_NODOS_ER, by itself, creates a new
COORDENADAS_NODOS_ER or raises the existing
%      singleton*.
%
%      H = COORDENADAS_NODOS_ER returns the handle to a new
COORDENADAS_NODOS_ER or the handle to
%      the existing singleton*.
%
%      COORDENADAS_NODOS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in COORDENADAS_NODOS_ER.M with the
given input arguments.
```

```
%
%     COORDENADAS_NODOS_ER('Property','Value',...) creates a new
COORDENADAS_NODOS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Coordenadas_Nodos_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Coordenadas_Nodos_ER_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Coordenadas_Nodos_ER

% Last Modified by GUIDE v2.5 21-Nov-2011 13:50:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Coordenadas_Nodos_ER_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Coordenadas_Nodos_ER_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Coordenadas_Nodos_ER is made visible.
function Coordenadas_Nodos_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Coordenadas_Nodos_ER (see
VARARGIN)

% Choose default command line output for Coordenadas_Nodos_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Coordenadas_Nodos_ER wait for user response (see
UIRESUME)
```

```
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Coordinadas_Nodos_ER_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorcoordnodret
%Se muestra en la ventana Coordinadas_Nodos_EA el valor de la variable
%contadorcoordnodret
set(handles.contadorcoordnodret_output, 'String', contadorcoordnodret)

function x_input_Callback(hObject, eventdata, handles)
% hObject      handle to x_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of x_input as text
%         str2double(get(hObject, 'String')) returns contents of x_input
%         as a double

% --- Executes during object creation, after setting all properties.
function x_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to x_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function y_input_Callback(hObject, eventdata, handles)
% hObject      handle to y_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of y_input as text
%         str2double(get(hObject, 'String')) returns contents of y_input
%         as a double

% --- Executes during object creation, after setting all properties.
function y_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to y_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)

global coordenadax coordenaday

%Se reocogen en las variables coordenadax y coordenaday los valores
%introducidos en las edit text de la ventanas Coordenadas_Nodos EA
coordenadax=str2double(get(handles.x_input,'String'));
coordenaday=str2double(get(handles.y_input,'String'));
%Se cierra la ventana
close Coordenadas_Nodos_ER

function contadorcoordinodret_output_Callback(hObject, eventdata,
handles)
% hObject    handle to contadorcoordinodret_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
contadorcoordinodret_output as text
% str2double(get(hObject,'String')) returns contents of
contadorcoordinodret_output as a double

% --- Executes during object creation, after setting all properties.
function contadorcoordinodret_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to contadorcoordinodret_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)

%Se cierra la ventana
close Coordenadas_Nodos_ER
```

### Coordenadas de cada nodo de la estructura articulada

```
function varargout = Coordenadas_Nodos_EA(varargin)
% COORDENADAS_NODOS_EA M-file for Coordenadas_Nodos_EA.fig
% COORDENADAS_NODOS_EA, by itself, creates a new
COORDENADAS_NODOS_EA or raises the existing
% singleton*.
```

```
%
%      H = COORDENADAS_NODOS_EA returns the handle to a new
COORDENADAS_NODOS_EA or the handle to
%      the existing singleton*.
%
%      COORDENADAS_NODOS_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in COORDENADAS_NODOS_EA.M with the
given input arguments.
%
%      COORDENADAS_NODOS_EA('Property','Value',...) creates a new
COORDENADAS_NODOS_EA or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Coordenadas_Nodos_EA_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to Coordenadas_Nodos_EA_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Coordenadas_Nodos_EA

% Last Modified by GUIDE v2.5 21-Nov-2011 14:31:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Coordenadas_Nodos_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Coordenadas_Nodos_EA_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Coordenadas_Nodos_EA is made visible.
function Coordenadas_Nodos_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to Coordenadas_Nodos_EA (see
VARARGIN)
```



```
% Choose default command line output for Coordenadas_Nodos_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Coordenadas_Nodos_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Coordenadas_Nodos_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numeronodocoordenadasart
%Se muestra en la ventana Coordenadas_Nodos_EA el valor de la variable
%numeronodocoordenadasart
set(handles.nodocoordart_output, 'String', numeronodocoordenadasart)

function x_input_Callback(hObject, eventdata, handles)
% hObject      handle to x_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of x_input as text
%         str2double(get(hObject, 'String')) returns contents of x_input
as a double

% --- Executes during object creation, after setting all properties.
function x_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to x_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function y_input_Callback(hObject, eventdata, handles)
% hObject      handle to y_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of y_input as text
%         str2double(get(hObject, 'String')) returns contents of y_input
as a double
```

```
% --- Executes during object creation, after setting all properties.
function y_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to y_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global coordenadax coordenaday

%Se reocogen en las variables coordenadax y coordenaday los valores
%introducidos en las edit text de la ventanas Coordinadas_Nodos EA
coordenadax=str2double(get(handles.x_input,'String'));
coordenaday=str2double(get(handles.y_input,'String'));

%Se cierra la ventana
close Coordinadas_Nodos_EA

function nodocoordart_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodocoordart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodocoordart_output
as text
%       str2double(get(hObject,'String')) returns contents of
nodocoordart_output as a double

% --- Executes during object creation, after setting all properties.
function nodocoordart_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodocoordart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana
close Coordenadas_Nodos_EA
```

## Tipo de apoyo de la estructura reticulada

```
function varargout = Tipo_Apoyo_ER(varargin)
% TIPO_APOYO_ER M-file for Tipo_Apoyo_ER.fig
%     TIPO_APOYO_ER, by itself, creates a new TIPO_APOYO_ER or raises
the existing
%     singleton*.
%
%     H = TIPO_APOYO_ER returns the handle to a new TIPO_APOYO_ER or
the handle to
%     the existing singleton*.
%
%     TIPO_APOYO_ER('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in TIPO_APOYO_ER.M with the given input
arguments.
%
%     TIPO_APOYO_ER('Property','Value',...) creates a new
TIPO_APOYO_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Tipo_Apoyo_ER_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Tipo_Apoyo_ER_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Tipo_Apoyo_ER

% Last Modified by GUIDE v2.5 21-Nov-2011 17:55:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Tipo_Apoyo_ER_OpeningFcn, ...
                  'gui_OutputFcn',  @Tipo_Apoyo_ER_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before Tipo_Apoyo_ER is made visible.
function Tipo_Apoyo_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Tipo_Apoyo_ER (see VARARGIN)

% Choose default command line output for Tipo_Apoyo_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Tipo_Apoyo_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Tipo_Apoyo_ER_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorrestret

%Se muestra en la ventana el valor de la variable contadorrestret
set(handles.contadorrestreti_output, 'String', contadorrestret)

function desph_input_Callback(hObject, eventdata, handles)
% hObject      handle to desph_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of desph_input as text
%         str2double(get(hObject, 'String')) returns contents of
desph_input as a double

% --- Executes during object creation, after setting all properties.
function desph_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to desph_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function despv_input_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to despv_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of despv_input as text
%         str2double(get(hObject,'String')) returns contents of
despv_input as a double

% --- Executes during object creation, after setting all properties.
function despv_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to despv_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global numnodorestringido desplazh desplazv giro

%Se asigna a las variables numnodorestringido desplazh desplazv giro
el
%valor introducido en las edit text de la ventana
numnodorestringido=str2double(get(handles.edit3,'String'));
desplazh=str2double(get(handles.desph_input,'String'));
desplazv=str2double(get(handles.despv_input,'String'));
giro=str2double(get(handles.giro_input,'String'));

%Se cierra la ventana
close Tipo_Apoyo_ER

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3
as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function giro_input_Callback(hObject, eventdata, handles)
% hObject      handle to giro_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of giro_input as text
%         str2double(get(hObject,'String')) returns contents of
giro_input as a double

% --- Executes during object creation, after setting all properties.
function giro_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to giro_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadorrestreti_output_Callback(hObject, eventdata, handles)
% hObject      handle to t (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of t as text
%         str2double(get(hObject,'String')) returns contents of t as a
double

% --- Executes during object creation, after setting all properties.
function t_CreateFcn(hObject, eventdata, handles)
% hObject      handle to t (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function contadorrestreti_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to contadorrestreti_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
```

```
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Tipo_Apoyo_ER
```

### Tipo de apoyo de la estructura articulada

```
function varargout = Tipo_Apoyo_EA(varargin)
% TIPO_APOYO_EA M-file for Tipo_Apoyo_EA.fig
%       TIPO_APOYO_EA, by itself, creates a new TIPO_APOYO_EA or raises
the existing
%       singleton*.
%
%       H = TIPO_APOYO_EA returns the handle to a new TIPO_APOYO_EA or
the handle to
%       the existing singleton*.
%
%       TIPO_APOYO_EA('CALLBACK',hObject,eventData,handles,...) calls
the local
%       function named CALLBACK in TIPO_APOYO_EA.M with the given input
arguments.
%
%       TIPO_APOYO_EA('Property','Value',...) creates a new
TIPO_APOYO_EA or raises the
%       existing singleton*. Starting from the left, property value
pairs are
%       applied to the GUI before Tipo_Apoyo_EA_OpeningFcn gets called.
An
%       unrecognized property name or invalid value makes property
application
%       stop. All inputs are passed to Tipo_Apoyo_EA_OpeningFcn via
varargin.
%
%       *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Tipo_Apoyo_EA

% Last Modified by GUIDE v2.5 21-Nov-2011 17:56:42

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Tipo_Apoyo_EA_OpeningFcn, ...
                  'gui_OutputFcn',  @Tipo_Apoyo_EA_OutputFcn, ...
```

```
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Tipo_Apoyo_EA is made visible.
function Tipo_Apoyo_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Tipo_Apoyo_EA (see VARARGIN)

% Choose default command line output for Tipo_Apoyo_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Tipo_Apoyo_EA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Tipo_Apoyo_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadornumrestriccart

%Se muestra en la ventana el valor de la variable contadorrestret
set(handles.nodotipapoyart_output,'String',contadornumrestriccart)

function desph_input_Callback(hObject, eventdata, handles)
% hObject    handle to desph_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of desph_input as text
%        str2double(get(hObject,'String')) returns contents of
desph_input as a double

% --- Executes during object creation, after setting all properties.
function desph_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to desph_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```



```
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function despv_input_Callback(hObject, eventdata, handles)
% hObject      handle to despv_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of despv_input as text
%      str2double(get(hObject,'String')) returns contents of
despv_input as a double

% --- Executes during object creation, after setting all properties.
function despv_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to despv_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numnodorestringido desplazh desplazv

%Se asigna a las variables numnodorestringido desplazh desplazv el
%valor introducido en las edit text de la ventana
numnodorestringido=str2double(get(handles.edit3,'String'));
desplazh=str2double(get(handles.desph_input,'String'));
desplazv=str2double(get(handles.despv_input,'String'));

%Se cierra la ventana
close Tipo_Apoyo_EA

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%      str2double(get(hObject,'String')) returns contents of edit3
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nodotipapoyart_output_Callback(hObject, eventdata, handles)
% hObject    handle to nodotipapoyart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nodotipapoyart_output as text
%       str2double(get(hObject,'String')) returns contents of
nodotipapoyart_output as a double

% --- Executes during object creation, after setting all properties.
function nodotipapoyart_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nodotipapoyart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Tipo_Apoyo_EA
```

### Valor de la constante de amortiguamiento de la estructura reticulada

```
function varargout = Valor_Cte_C_ER(varargin)
% VALOR_CTE_C_ER M-file for Valor_Cte_C_ER.fig
%       VALOR_CTE_C_ER, by itself, creates a new VALOR_CTE_C_ER or
raises the existing
%       singleton*.
%
%       H = VALOR_CTE_C_ER returns the handle to a new VALOR_CTE_C_ER
or the handle to
```

```
% the existing singleton*.
%
% VALOR_CTE_C_ER('CALLBACK',hObject,eventData,handles,...) calls
the local
% function named CALLBACK in VALOR_CTE_C_ER.M with the given
input arguments.
%
% VALOR_CTE_C_ER('Property','Value',...) creates a new
VALOR_CTE_C_ER or raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before Valor_Cte_C_ER_OpeningFcn gets
called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to Valor_Cte_C_ER_OpeningFcn via
varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Valor_Cte_C_ER

% Last Modified by GUIDE v2.5 21-Nov-2011 19:12:30

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Valor_Cte_C_ER_OpeningFcn, ...
                  'gui_OutputFcn',  @Valor_Cte_C_ER_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Valor_Cte_C_ER is made visible.
function Valor_Cte_C_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valor_Cte_C_ER (see VARARGIN)

% Choose default command line output for Valor_Cte_C_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes Valor_Cte_C_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Valor_Cte_C_ER_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadoramorret

%Mostrar en la ventana el valor asignado a la variable contadoramorret
set(handles.contadoramorret_output, 'String', contadoramorret)

function ctec_input_Callback(hObject, eventdata, handles)
% hObject      handle to ctec_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of ctec_input as text
%         str2double(get(hObject, 'String')) returns contents of
%         ctec_input as a double

% --- Executes during object creation, after setting all properties.
function ctec_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ctec_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global valordec numerodeamortiguador

%Se asignan los valores introducidos en los edit text de la ventana a
las
%variables numerodeamortiguador y valordec
numerodeamortiguador=str2double(get(handles.numerodeamortiguador_input
, 'String'));
valordec=str2double(get(handles.ctec_input, 'String'));

%Se cierra la ventana
close Valor_Cte_C_ER
```

```
function numerodeamortiguador_input_Callback(hObject, eventdata, handles)
% hObject      handle to numerodeamortiguador_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numerodeamortiguador_input as text
%          str2double(get(hObject,'String')) returns contents of
numerodeamortiguador_input as a double

% --- Executes during object creation, after setting all properties.
function numerodeamortiguador_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numerodeamortiguador_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadoramorret_output_Callback(hObject, eventdata, handles)
% hObject      handle to contadoramorret_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
contadoramorret_output as text
%          str2double(get(hObject,'String')) returns contents of
contadoramorret_output as a double

% --- Executes during object creation, after setting all properties.
function contadoramorret_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to contadoramorret_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Valor_Cte_C_ER
```

## Valor de la constante de amortiguamiento de la estructura articulada

```
function varargout = Valor_Cte_C_EA(varargin)
% VALOR_CTE_C_EA M-file for Valor_Cte_C_EA.fig
%     VALOR_CTE_C_EA, by itself, creates a new VALOR_CTE_C_EA or
raises the existing
%     singleton*.
%
%     H = VALOR_CTE_C_EA returns the handle to a new VALOR_CTE_C_EA
or the handle to
%     the existing singleton*.
%
%     VALOR_CTE_C_EA('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in VALOR_CTE_C_EA.M with the given
input arguments.
%
%     VALOR_CTE_C_EA('Property','Value',...) creates a new
VALOR_CTE_C_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Valor_Cte_C_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Valor_Cte_C_EA_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Valor_Cte_C_EA

% Last Modified by GUIDE v2.5 21-Nov-2011 19:12:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Valor_Cte_C_EA_OpeningFcn, ...
                  'gui_OutputFcn',  @Valor_Cte_C_EA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Valor_Cte_C_EA is made visible.
function Valor_Cte_C_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valor_Cte_C_EA (see VARARGIN)

% Choose default command line output for Valor_Cte_C_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Valor_Cte_C_EA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Valor_Cte_C_EA_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorbaramoart

%Mostrar en la ventana el valor asignado a la variable
contadorbaramoart
set(handles.barracontadorbaramoart_output, 'String', contadorbaramoart)

function ctec_input_Callback(hObject, eventdata, handles)
% hObject    handle to ctec_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of ctec_input as text
%         str2double(get(hObject, 'String')) returns contents of
%         ctec_input as a double

% --- Executes during object creation, after setting all properties.
function ctec_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ctec_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
%             called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global valordec numerodeamortiguador
```

```
%Se asignan los valores introducidos en los edit text de la ventana a
las
%variables numerodeamortiguador y valordec
numerodeamortiguador=str2double(get(handles.numerodeamortiguador_input
,'String'));
valordec=str2double(get(handles.ctec_input,'String'));

%Se cierra la ventana
close Valor_Cte_C_EA
```

```
function numerodeamortiguador_input_Callback(hObject, eventdata,
handles)
% hObject    handle to numerodeamortiguador_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numerodeamortiguador_input as text
%         str2double(get(hObject,'String')) returns contents of
numerodeamortiguador_input as a double

% --- Executes during object creation, after setting all properties.
function numerodeamortiguador_input_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to numerodeamortiguador_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function barracontadorbaramoart_output_Callback(hObject, eventdata,
handles)
% hObject    handle to barracontadorbaramoart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
barracontadorbaramoart_output as text
%         str2double(get(hObject,'String')) returns contents of
barracontadorbaramoart_output as a double

% --- Executes during object creation, after setting all properties.
function barracontadorbaramoart_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to barracontadorbaramoart_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```



```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Se cierra la ventana
close Valor_Cte_C_EA
```

### Especificación de las características de las secciones de la estructura reticulada

```
function varargout = Especificacion_Secciones_ER(varargin)
% ESPECIFICACION_SECCIONES_ER M-file for
% Especificacion_Secciones_ER.fig
%   ESPECIFICACION_SECCIONES_ER, by itself, creates a new
%   ESPECIFICACION_SECCIONES_ER or raises the existing
%   singleton*.
%
%   H = ESPECIFICACION_SECCIONES_ER returns the handle to a new
%   ESPECIFICACION_SECCIONES_ER or the handle to
%   the existing singleton*.
%
%   ESPECIFICACION_SECCIONES_ER('CALLBACK',hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in ESPECIFICACION_SECCIONES_ER.M with
%   the given input arguments.
%
%   ESPECIFICACION_SECCIONES_ER('Property','Value',...) creates a
%   new ESPECIFICACION_SECCIONES_ER or raises the
%   existing singleton*. Starting from the left, property value
%   pairs are
%   applied to the GUI before
%   Especificacion_Secciones_ER_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to
%   Especificacion_Secciones_ER_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%   only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Especificacion_Secciones_ER

% Last Modified by GUIDE v2.5 18-Nov-2011 19:39:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',
```

```
@Especificacion_Secciones_ER_OpeningFcn, ...
    'gui_OutputFcn',
@Especificacion_Secciones_ER_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Especificacion_Secciones_ER is made
% visible.
function Especificacion_Secciones_ER_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Especificacion_Secciones_ER
% (see VARARGIN)

% Choose default command line output for Especificacion_Secciones_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Especificacion_Secciones_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Especificacion_Secciones_ER_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_materiales tablamateriales

%se muestran en la ventana los valores de las variables
numero_materiales y
tablamateriales
set(handles.nummaterial_output,'String',numero_materiales)
set(handles.uitable1,'data',tablamateriales)

% --- Executes on button press in tipos_material.
function tipos_material_Callback(hObject, eventdata, handles)
% hObject    handle to tipos_material (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana Tipos_Seccion_ER
Tipos_Seccion_ER

uiwait

global numero_materiales
%Se muestra en la ventana el valor de la variable numero_materiales
set(handles.nummaterial_output, 'String', numero_materiales)

% --- Executes on button press in asignacion_material.
function asignacion_material_Callback(hObject, eventdata, handles)
% hObject      handle to asignacion_material (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_barras seccion A E I densidad
barracolespecificacionmateriales numero_materiales contadorasigsecret

%se obtienen los vectores de E A I y densidad con los valores del
modulo
%elastico densidad momento de inercia y area de cada barra
if numero_materiales==1
    for i=1:numero_barras
        E(i,:)=seccion(1,1);
    A(i,:)=seccion(1,2);
    I(i,:)=seccion(1,3);
    densidad(i,:)=seccion(1,4);
    barracolespecificacionmateriales(i,:)=i;
    end
elseif numero_materiales>1

    for i=1:numero_barras
        contadorasigsecret=i;
    Asignacion_Seccion

uiwait

global asignacionbarra asignacionseccion
E(asignacionbarra,:)=seccion(asignacionseccion,1);
A(asignacionbarra,:)=seccion(asignacionseccion,2);
I(asignacionbarra,:)=seccion(asignacionseccion,3);
densidad(asignacionbarra,:)=seccion(asignacionseccion,4);
barracolespecificacionmateriales(i,:)=i;
    end

end

% --- Executes on button press in especificacion_material.
function especificacion_material_Callback(hObject, eventdata, handles)
% hObject      handle to especificacion_material (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_materiales seccion contadorespecificsecret
tablamateriales

%Se obtiene una matriz que contiene los valores de los tipos de
secciones
%que tiene la estructura, estos valores seran el modulo de
elasticidad,
```

```
%momento de inercia densidad y area
for i=1:numero_materiales
    contadorespecificseccret=i;
    Caracteristicas_de_Seccion

    uiwait

    global Ee Aa Ii densidadd
    seccion(i,:)=[Ee Aa Ii densidadd];
    Eee(i,:)=Ee;
    Aaa(i,:)=Aa;
    Iii(i,:)=Ii;
    densidaddd(i,:)=densidadd;
    fuercolvectorreticulada=[1:1:i];
    fuercolreticulada=fuercolvectorreticulada';

end

tablamateriales=[fuercolreticulada Eee Aaa Iii densidaddd];

%se muestra en la ventana los valores de la matriz tablamateriales

set(handles.uitable1,'data',tablamateriales)


function nummaterial_output_Callback(hObject, eventdata, handles)
% hObject      handle to nummaterial_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nummaterial_output
as text
%      str2double(get(hObject,'String')) returns contents of
nummaterial_output as a double

% --- Executes during object creation, after setting all properties.
function nummaterial_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nummaterial_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function matdensidad_output_Callback(hObject, eventdata, handles)
% hObject      handle to matdensidad_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of matdensidad_output
as text
%      str2double(get(hObject,'String')) returns contents of
matdensidad_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function matdensidad_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to matdensidad_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function matI_output_Callback(hObject, eventdata, handles)
% hObject      handle to matI_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of matI_output as text
%         str2double(get(hObject,'String')) returns contents of
matI_output as a double

% --- Executes during object creation, after setting all properties.
function matI_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to matI_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function matA_output_Callback(hObject, eventdata, handles)
% hObject      handle to matA_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of matA_output as text
%         str2double(get(hObject,'String')) returns contents of
matA_output as a double

% --- Executes during object creation, after setting all properties.
function matA_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to matA_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function matE_output_Callback(hObject, eventdata, handles)
% hObject      handle to matE_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of matE_output as text
%         str2double(get(hObject,'String')) returns contents of
matE_output as a double

% --- Executes during object creation, after setting all properties.
function matE_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to matE_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close Especificacion_Secciones_ER

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of edit7
as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit8_Callback(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of edit8
as a double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9
as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%         str2double(get(hObject,'String')) returns contents of edit10
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nummatbar_output_Callback(hObject, eventdata, handles)
% hObject    handle to nummatbar_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nummatbar_output as
text
%       str2double(get(hObject,'String')) returns contents of
nummatbar_output as a double

% --- Executes during object creation, after setting all properties.
function nummatbar_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nummatbar_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%       str2double(get(hObject,'String')) returns contents of edit12
as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```



```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

### Especificación de las características de las secciones de la estructura articulada

```
function varargout = Especificacion_Secciones(varargin)  
% ESPECIFICACION_SECCIONES M-file for Especificacion_Secciones.fig  
%     ESPECIFICACION_SECCIONES, by itself, creates a new  
% ESPECIFICACION_SECCIONES or raises the existing  
%     singleton*.  
%  
%     H = ESPECIFICACION_SECCIONES returns the handle to a new  
% ESPECIFICACION_SECCIONES or the handle to  
%     the existing singleton*.  
%  
%  
% ESPECIFICACION_SECCIONES('CALLBACK',hObject,eventData,handles,...)  
% calls the local  
%     function named CALLBACK in ESPECIFICACION_SECCIONES.M with the  
% given input arguments.  
%  
%     ESPECIFICACION_SECCIONES('Property','Value',...) creates a new  
% ESPECIFICACION_SECCIONES or raises the  
%     existing singleton*. Starting from the left, property value  
% pairs are  
%     applied to the GUI before Especificacion_Secciones_OpeningFcn  
% gets called. An  
%     unrecognized property name or invalid value makes property  
% application  
%     stop. All inputs are passed to  
% Especificacion_Secciones_OpeningFcn via varargin.  
%  
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
% only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help  
% Especificacion_Secciones  
  
% Last Modified by GUIDE v2.5 06-Nov-2011 17:19:56  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',  gui_Singleton, ...  
                  'gui_OpeningFcn', @Especificacion_Secciones_OpeningFcn, ...  
                  'gui_OutputFcn',  @Especificacion_Secciones_OutputFcn, ...  
                  'gui_LayoutFcn',  [], ...  
                  'gui_Callback',   []);  
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end  
  
if nargout  
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});  
end
```

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Especificacion_Secciones is made visible.
function Especificacion_Secciones_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Especificacion_Secciones (see
VARARGIN)

% Choose default command line output for Especificacion_Secciones
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Especificacion_Secciones wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Especificacion_Secciones_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_secciones tablacaract

%se muestran en la ventana los valores de las variables
numero_secciones y
%tablacaract
set(handles.numseccion_output, 'String', numero_secciones)
set(handles.uitable1, 'data', tablacaract)

% --- Executes on button press in tipos_seccion.
function tipos_seccion_Callback(hObject, eventdata, handles)
% hObject      handle to tipos_seccion (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)
%Se abre la ventana Tipos_de_Seccion
Tipos_de_Seccion

uiwait

global numero_secciones
%Se muestra en la ventana el valor de la variable numero_secciones
set(handles.numseccion_output, 'String', numero_secciones)

% --- Executes on button press in asignacion_seccion.
function asignacion_seccion_Callback(hObject, eventdata, handles)
% hObject      handle to asignacion_seccion (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global numero_barras seccion A E densidad
barracolespecificacionsecciones numero_secciones contadorbarseccart

%se obtienen los vectores de E A y densidad con los valores del modulo
%elastico densidad momento de inercia y area de cada barra
if numero_secciones==1
    for i=1:numero_barras
        E(i,:)=seccion(1,1);
        A(i,:)=seccion(1,2);

        densidad(i,:)=seccion(1,3);
        barracolespecificacionsecciones(i,:)=i;
    end

elseif numero_secciones>1

    for i=1:numero_barras
        contadorbarseccart=i;
        Asignacion_Seccion_Articuladas

        uiwait

        global asignacionbarra asignacionseccion
        E(asignacionbarra,:)=seccion(asignacionseccion,1);
        A(asignacionbarra,:)=seccion(asignacionseccion,2);

        densidad(asignacionbarra,:)=seccion(asignacionseccion,3);
        barracolespecificacionsecciones(i,:)=i;
    end

end

% --- Executes on button press in especificacion_seccion.
function especificacion_seccion_Callback(hObject, eventdata, handles)
% hObject handle to especificacion_seccion (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global numero_secciones seccion tablacaract

%Se obtiene una matriz que contiene los valores de los tipos de
secciones
%que tiene la estructura, estos valores seran el modulo de
elasticidad,
% densidad y area
for i=1:numero_secciones
    Caracteristicas_de_Seccion_Articuladas

    uiwait

    global Ee Aa Ii densidadd
    seccion(i,:)=[Ee Aa Ii densidadd];
    Ee(i,:)=Ee;
    Aa(i,:)=Aa;

    densidadd(i,:)=densidadd;
    fuercolvectorarticulada=[1:1:i];
    fuercolarticulada=fuercolvectorarticulada';
```

end

```
tablacaract=[fuercolarticulada Eee Aaa densidaddd];  
%se muestra en la ventana los valores de la matriz tablacaract  
set(handles.uitable1,'data',tablacaract)
```

```
function numseccion_output_Callback(hObject, eventdata, handles)  
% hObject    handle to numseccion_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of numseccion_output  
as text  
%         str2double(get(hObject,'String')) returns contents of  
numseccion_output as a double
```

```
% --- Executes during object creation, after setting all properties.  
function numseccion_output_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to numseccion_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function seccdensidad_output_Callback(hObject, eventdata, handles)  
% hObject    handle to seccdensidad_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of seccdensidad_output  
as text  
%         str2double(get(hObject,'String')) returns contents of  
seccdensidad_output as a double
```

```
% --- Executes during object creation, after setting all properties.  
function seccdensidad_output_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to seccdensidad_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function seccA_output_Callback(hObject, eventdata, handles)  
% hObject    handle to seccA_output (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of seccA_output as
text
%          str2double(get(hObject,'String')) returns contents of
seccA_output as a double

% --- Executes during object creation, after setting all properties.
function seccA_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to seccA_output (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function seccE_output_Callback(hObject, eventdata, handles)
% hObject      handle to seccE_output (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of seccE_output as
text
%          str2double(get(hObject,'String')) returns contents of
seccE_output as a double

% --- Executes during object creation, after setting all properties.
function seccE_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to seccE_output (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close Especificacion_Secciones

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6
%         as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of edit7
%         as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9
%         as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
```

```
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numsecbar_output_Callback(hObject, eventdata, handles)
% hObject      handle to numsecbar_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numsecbar_output
as text
%      str2double(get(hObject,'String')) returns contents of
numsecbar_output as a double

% --- Executes during object creation, after setting all properties.
function numsecbar_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numsecbar_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%      str2double(get(hObject,'String')) returns contents of edit11
as a double

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

## Número de barras de la estructura reticulada

```
function varargout = Numero_Barras_ER(varargin)
% NUMERO_BARRAS_ER M-file for Numero_Barras_ER.fig
%     NUMERO_BARRAS_ER, by itself, creates a new NUMERO_BARRAS_ER or
raises the existing
%     singleton*.
%
%     H = NUMERO_BARRAS_ER returns the handle to a new
NUMERO_BARRAS_ER or the handle to
%     the existing singleton*.
%
%     NUMERO_BARRAS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_BARRAS_ER.M with the given
input arguments.
%
%     NUMERO_BARRAS_ER('Property','Value',...) creates a new
NUMERO_BARRAS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Barras_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_Barras_ER_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Barras_ER

% Last Modified by GUIDE v2.5 18-Nov-2011 18:59:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Barras_ER_OpeningFcn, ...
                  'gui_OutputFcn',  @Numero_Barras_ER_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Barras_ER is made visible.
function Numero_Barras_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
```



```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Barras_ER (see VARARGIN)

% Choose default command line output for Numero_Barras_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Barras_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Barras_ER_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Numero_Barras_ER

function numbarras_input_Callback(hObject, eventdata, handles)
% hObject    handle to numbarras_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numbarras_input as
text
%         str2double(get(hObject,'String')) returns contents of
numbarras_input as a double

% --- Executes during object creation, after setting all properties.
function numbarras_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numbarras_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numbarras_output_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to numbarras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numbarras_output as
text
%          str2double(get(hObject,'String')) returns contents of
numbarras_output as a double

% --- Executes during object creation, after setting all properties.
function numbarras_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numbarras_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_barras
%Se asigna el valor introducido en la edit text de la ventana a la
variable
%numero_barras
numero_barras=str2double(get(handles.numbarras_input,'String'));
%se muestra en la ventana el valor de la variable numero_barras
set(handles.numbarras_output,'String',numero_barras);
```

### Número de barras de la estructura articulada

```
function varargout = Numero_Barras_EA(varargin)
% NUMERO_BARRAS_EA M-file for Numero_Barras_EA.fig
%     NUMERO_BARRAS_EA, by itself, creates a new NUMERO_BARRAS_EA or
raises the existing
%     singleton*.
%
%     H = NUMERO_BARRAS_EA returns the handle to a new
NUMERO_BARRAS_EA or the handle to
%     the existing singleton*.
%
%     NUMERO_BARRAS_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_BARRAS_EA.M with the given
input arguments.
%
%     NUMERO_BARRAS_EA('Property','Value',...) creates a new
NUMERO_BARRAS_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Barras_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
```

```
application
%      stop. All inputs are passed to Numero_Barras_EA_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Barras_EA

% Last Modified by GUIDE v2.5 18-Nov-2011 18:58:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Barras_EA_OpeningFcn, ...
                  'gui_OutputFcn',  @Numero_Barras_EA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Barras_EA is made visible.
function Numero_Barras_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Barras_EA (see VARARGIN)

% Choose default command line output for Numero_Barras_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Barras_EA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Barras_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Numero_Barras_EA

function numbarras_input_Callback(hObject, eventdata, handles)
% hObject      handle to numbarras_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numbarras_input as
text
%          str2double(get(hObject,'String')) returns contents of
numbarras_input as a double

% --- Executes during object creation, after setting all properties.
function numbarras_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numbarras_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numbarras_output_Callback(hObject, eventdata, handles)
% hObject      handle to numbarras_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numbarras_output as
text
%          str2double(get(hObject,'String')) returns contents of
numbarras_output as a double

% --- Executes during object creation, after setting all properties.
function numbarras_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numbarras_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_barras
%Se asigna el valor introducido en la edit text de la ventana a la
variable
%numero_barras
numero_barras=str2double(get(handles.numbarras_input,'String'));
%Se muestra en pantalla el valor de la variable numero_barras
set(handles.numbarras_output,'String',numero_barras);
```

### Número de apoyos de la estructura reticulada

```
function varargout = Numero_Nodos_Restringidos_ER(varargin)
% NUMERO_NODOS_RESTRINGIDOS_ER M-file for
Numero_Nodos_Restringidos_ER.fig
%     NUMERO_NODOS_RESTRINGIDOS_ER, by itself, creates a new
NUMERO_NODOS_RESTRINGIDOS_ER or raises the existing
%     singleton*.
%
%     H = NUMERO_NODOS_RESTRINGIDOS_ER returns the handle to a new
NUMERO_NODOS_RESTRINGIDOS_ER or the handle to
%     the existing singleton*.
%
%
NUMERO_NODOS_RESTRINGIDOS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_NODOS_RESTRINGIDOS_ER.M with
the given input arguments.
%
%     NUMERO_NODOS_RESTRINGIDOS_ER('Property','Value',...) creates a
new NUMERO_NODOS_RESTRINGIDOS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Numero_Nodos_Restringidos_ER_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Numero_Nodos_Restringidos_ER_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Numero_Nodos_Restringidos_ER

% Last Modified by GUIDE v2.5 18-Nov-2011 19:07:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Nodos_Restringidos_ER_OpeningFcn, ...
                  'gui_OutputFcn',
```

```
@Numero_Nodos_Restringidos_ER_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Nodos_Restringidos_ER is made
% visible.
function Numero_Nodos_Restringidos_ER_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Nodos_Restringidos_ER
% (see VARARGIN)

% Choose default command line output for Numero_Nodos_Restringidos_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Nodos_Restringidos_ER wait for user response
% (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Nodos_Restringidos_ER_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Numero_Nodos_Restringidos_ER

function numrestricc_input_Callback(hObject, eventdata, handles)
% hObject    handle to numrestricc_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of numrestricc_input
as text
%         str2double(get(hObject,'String')) returns contents of
numrestricc_input as a double

% --- Executes during object creation, after setting all properties.
function numrestricc_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numrestricc_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numrestricc_output_Callback(hObject, eventdata, handles)
% hObject    handle to numrestricc_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numrestricc_output
as text
%         str2double(get(hObject,'String')) returns contents of
numrestricc_output as a double

% --- Executes during object creation, after setting all properties.
function numrestricc_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numrestricc_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_restricciones

%Se asigna el valor introducido en la edit text a la variable
%numero_restricciones
numero_restricciones=str2double(get(handles.numrestricc_input,'String'
));
%se muestra en pantalla el valor de la variables numero_restricciones
set(handles.numrestricc_output,'String',numero_restricciones);
```

## Número de apoyos de la estructura articulada

```
function varargout = Numero_Nodos_Restringidos_EA(varargin)
% NUMERO_NODOS_RESTRINGIDOS_EA M-file for
Numero_Nodos_Restringidos_EA.fig
%     NUMERO_NODOS_RESTRINGIDOS_EA, by itself, creates a new
NUMERO_NODOS_RESTRINGIDOS_EA or raises the existing
%     singleton*.
%
%     H = NUMERO_NODOS_RESTRINGIDOS_EA returns the handle to a new
NUMERO_NODOS_RESTRINGIDOS_EA or the handle to
%     the existing singleton*.
%
%
NUMERO_NODOS_RESTRINGIDOS_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_NODOS_RESTRINGIDOS_EA.M with
the given input arguments.
%
%     NUMERO_NODOS_RESTRINGIDOS_EA('Property','Value',...) creates a
new NUMERO_NODOS_RESTRINGIDOS_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Numero_Nodos_Restringidos_EA_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Numero_Nodos_Restringidos_EA_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Numero_Nodos_Restringidos_EA

% Last Modified by GUIDE v2.5 18-Nov-2011 19:06:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Nodos_Restringidos_EA_OpeningFcn, ...
                  'gui_OutputFcn',  @Numero_Nodos_Restringidos_EA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```



```
% --- Executes just before Numero_Nodos_Restringidos_EA is made
visible.
function Numero_Nodos_Restringidos_EA_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Nodos_Restringidos_EA
(see VARARGIN)

% Choose default command line output for Numero_Nodos_Restringidos_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Nodos_Restringidos_EA wait for user response
(see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Nodos_Restringidos_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Numero_Nodos_Restringidos_EA

function numrestricc_input_Callback(hObject, eventdata, handles)
% hObject      handle to numrestricc_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numrestricc_input
as text
%          str2double(get(hObject,'String')) returns contents of
numrestricc_input as a double

% --- Executes during object creation, after setting all properties.
function numrestricc_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numrestricc_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numrestricc_output_Callback(hObject, eventdata, handles)
% hObject    handle to numrestricc_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numrestricc_output
as text
% str2double(get(hObject,'String')) returns contents of
numrestricc_output as a double

% --- Executes during object creation, after setting all properties.
function numrestricc_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numrestricc_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_restricciones

%Se asigna el valor introducido en la edit text a la variable
numero_restricciones
numero_restricciones=str2double(get(handles.numrestricc_input,'String'
));
%se muestra en pantalla el valor de la variables numero_restricciones
set(handles.numrestricc_output,'String',numero_restricciones);
```

### Número de muelles que tiene la estructura reticulada

```
function varargout = Numero_Muelles_ER(varargin)
% NUMERO_MUELLES_ER M-file for Numero_Muelles_ER.fig
% NUMERO_MUELLES_ER, by itself, creates a new NUMERO_MUELLES_ER
or raises the existing
% singleton*.
%
% H = NUMERO_MUELLES_ER returns the handle to a new
NUMERO_MUELLES_ER or the handle to
% the existing singleton*.
%
% NUMERO_MUELLES_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
```

```
%      function named CALLBACK in NUMERO_MUELLES_ER.M with the given
input arguments.
%
%      NUMERO_MUELLES_ER('Property','Value',...) creates a new
NUMERO_MUELLES_ER or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Numero_Muelles_ER_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to Numero_Muelles_ER_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Muelles_ER

% Last Modified by GUIDE v2.5 18-Nov-2011 19:14:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Muelles_ER_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Numero_Muelles_ER_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Muelles_ER is made visible.
function Numero_Muelles_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to Numero_Muelles_ER (see
VARARGIN)

% Choose default command line output for Numero_Muelles_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Muelles_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = Numero_Muelles_ER_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject     handle to cerrar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Cerrar la ventana
close Numero_Muelles_ER

function numuelles_input_Callback(hObject, eventdata, handles)
% hObject     handle to numuelles_input (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numuelles_input as
text
%          str2double(get(hObject,'String')) returns contents of
numuelles_input as a double

% --- Executes during object creation, after setting all properties.
function numuelles_input_CreateFcn(hObject, eventdata, handles)
% hObject     handle to numuelles_input (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numuelles_output_Callback(hObject, eventdata, handles)
% hObject     handle to numuelles_output (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numuelles_output as
text
%          str2double(get(hObject,'String')) returns contents of
numuelles_output as a double

% --- Executes during object creation, after setting all properties.
```

```
function numuelles_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numuelles_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_muelles
%Se asigna el valor introducido en la edit text a la variable
%numero_muelles
numero_muelles=str2double(get(handles.numuelles_input,'String'));
%Se muestra en pantalla el valor de la variable numero_muelles
set(handles.numuelles_output,'String',numero_muelles);
```

### Número de muelles que tiene la estructura articulada

```
function varargout = Numero_Muelles_EA(varargin)
% NUMERO_MUELLES_EA M-file for Numero_Muelles_EA.fig
%     NUMERO_MUELLES_EA, by itself, creates a new NUMERO_MUELLES_EA
or raises the existing
%     singleton*.
%
%     H = NUMERO_MUELLES_EA returns the handle to a new
NUMERO_MUELLES_EA or the handle to
%     the existing singleton*.
%
%     NUMERO_MUELLES_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_MUELLES_EA.M with the given
input arguments.
%
%     NUMERO_MUELLES_EA('Property','Value',...) creates a new
NUMERO_MUELLES_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Muelles_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_Muelles_EA_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Numero_Muelles_EA
```

```
% Last Modified by GUIDE v2.5 18-Nov-2011 19:13:17

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_Muelles_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',   @Numero_Muelles_EA_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Numero_Muelles_EA is made visible.
function Numero_Muelles_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Muelles_EA (see
VARARGIN)

% Choose default command line output for Numero_Muelles_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Muelles_EA wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Numero_Muelles_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Se cierra la ventana
close Numero_Muelles_EA
```

```
function numuelles_input_Callback(hObject, eventdata, handles)
% hObject      handle to numuelles_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numuelles_input as
text
%          str2double(get(hObject,'String')) returns contents of
numuelles_input as a double

% --- Executes during object creation, after setting all properties.
function numuelles_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numuelles_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numuelles_output_Callback(hObject, eventdata, handles)
% hObject      handle to numuelles_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numuelles_output as
text
%          str2double(get(hObject,'String')) returns contents of
numuelles_output as a double

% --- Executes during object creation, after setting all properties.
function numuelles_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numuelles_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_muelles
%Se asigna el valor introducido en la edit text a la variable
%numero_muelles
```

```
numero_muelles=str2double(get(handles.numuelles_input,'String'));  
%Se muestra en pantalla el valor de la variable numero_muelles  
set(handles.numuelles_output,'String',numero_muelles);
```

### Número de amortiguadores que tiene la estructura reticulada

```
function varargout = Numero_Amortiguadores_ER(varargin)  
% NUMERO_AMORTIGUADORES_ER M-file for Numero_Amortiguadores_ER.fig  
%     NUMERO_AMORTIGUADORES_ER, by itself, creates a new  
NUMERO_AMORTIGUADORES_ER or raises the existing  
%     singleton*.  
%  
%     H = NUMERO_AMORTIGUADORES_ER returns the handle to a new  
NUMERO_AMORTIGUADORES_ER or the handle to  
%     the existing singleton*.  
%  
%  
NUMERO_AMORTIGUADORES_ER('CALLBACK',hObject,eventData,handles,...)  
calls the local  
%     function named CALLBACK in NUMERO_AMORTIGUADORES_ER.M with the  
given input arguments.  
%  
%     NUMERO_AMORTIGUADORES_ER('Property','Value',...) creates a new  
NUMERO_AMORTIGUADORES_ER or raises the  
%     existing singleton*. Starting from the left, property value  
pairs are  
%     applied to the GUI before Numero_Amortiguadores_ER_OpeningFcn  
gets called. An  
%     unrecognized property name or invalid value makes property  
application  
%     stop. All inputs are passed to  
Numero_Amortiguadores_ER_OpeningFcn via varargin.  
%  
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help  
Numero_Amortiguadores_ER  
  
% Last Modified by GUIDE v2.5 18-Nov-2011 19:24:30  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',   gui_Singleton, ...  
                  'gui_OpeningFcn',  @Numero_Amortiguadores_ER_OpeningFcn, ...  
                  'gui_OutputFcn',   @Numero_Amortiguadores_ER_OutputFcn, ...  
                  'gui_LayoutFcn',   [], ...  
                  'gui_Callback',    []);  
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end  
  
if nargin  
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});  
else  
    gui_mainfcn(gui_State, varargin{:});  
end
```



```
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Amortiguadores_ER is made visible.
function Numero_Amortiguadores_ER_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Amortiguadores_ER (see VARARGIN)

% Choose default command line output for Numero_Amortiguadores_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Amortiguadores_ER wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Amortiguadores_ER_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Se cierra la ventana
close Numero_Amortiguadores_ER

function nuamortiguadores_input_Callback(hObject, eventdata, handles)
% hObject    handle to nuamortiguadores_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
%         nuamortiguadores_input as text
%         str2double(get(hObject,'String')) returns contents of
%         nuamortiguadores_input as a double

% --- Executes during object creation, after setting all properties.
function nuamortiguadores_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nuamortiguadores_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nuamortiguadores_output_Callback(hObject, eventdata, handles)
% hObject    handle to nuamortiguadores_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nuamortiguadores_output as text
%     str2double(get(hObject,'String')) returns contents of
nuamortiguadores_output as a double

% --- Executes during object creation, after setting all properties.
function nuamortiguadores_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to nuamortiguadores_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_amortiguadores
%se asigna el valor introducido en la edit text a la variables
numero_amortiguadores
numero_amortiguadores=str2double(get(handles.nuamortiguadores_input,'S
tring'));
%Se muestra en pantalla el valor de la variable numero_amortiguadores
set(handles.nuamortiguadores_output,'String',numero_amortiguadores);
```

### Número de amortiguadores que tiene la estructura articulada

```
function varargout = Numero_Amortiguadores_EA(varargin)
% NUMERO_AMORTIGUADORES_EA M-file for Numero_Amortiguadores_EA.fig
%     NUMERO_AMORTIGUADORES_EA, by itself, creates a new
NUMERO_AMORTIGUADORES_EA or raises the existing
%     singleton*.
%
%     H = NUMERO_AMORTIGUADORES_EA returns the handle to a new
NUMERO_AMORTIGUADORES_EA or the handle to
%     the existing singleton*.
%
```

```
%
NUMERO_AMORTIGUADORES_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_AMORTIGUADORES_EA.M with the
given input arguments.
%
%     NUMERO_AMORTIGUADORES_EA('Property','Value',...) creates a new
NUMERO_AMORTIGUADORES_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Amortiguadores_EA_OpeningFcn
gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Numero_Amortiguadores_EA_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Numero_Amortiguadores_EA

% Last Modified by GUIDE v2.5 18-Nov-2011 19:23:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Numero_Amortiguadores_EA_OpeningFcn, ...
                  'gui_OutputFcn',   @Numero_Amortiguadores_EA_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Amortiguadores_EA is made visible.
function Numero_Amortiguadores_EA_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Amortiguadores_EA (see
VARARGIN)

% Choose default command line output for Numero_Amortiguadores_EA
handles.output = hObject;
```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Amortiguadores_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Amortiguadores_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana
close Numero_Amortiguadores_EA

function nuamortiguadores_input_Callback(hObject, eventdata, handles)
% hObject      handle to nuamortiguadores_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nuamortiguadores_input as text
%          str2double(get(hObject,'String')) returns contents of
nuamortiguadores_input as a double

% --- Executes during object creation, after setting all properties.
function nuamortiguadores_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nuamortiguadores_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nuamortiguadores_output_Callback(hObject, eventdata, handles)
% hObject      handle to nuamortiguadores_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nuamortiguadores_output as text
```

```
%      str2double(get(hObject,'String')) returns contents of
nuamortiguadores_output as a double

% --- Executes during object creation, after setting all properties.
function nuamortiguadores_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to nuamortiguadores_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_amortiguadores
%se asigna el valor introducido en la edit text a la variables
%numero_amortiguadores
numero_amortiguadores=str2double(get(handles.nuamortiguadores_input,'S
tring'));
%Se muestra en pantalla el valor de la variable numero_amortiguadores
set(handles.nuamortiguadores_output,'String',numero_amortiguadores);
```

### Definición de fuerzas estáticas para la estructura reticulada

```
function varargout = Fuerza_Estatico_Reticulada(varargin)
% FUERZA_ESTATICO_RETICULADA M-file for Fuerza_Estatico_Reticulada.fig
%      FUERZA_ESTATICO_RETICULADA, by itself, creates a new
FUERZA_ESTATICO_RETICULADA or raises the existing
%      singleton*.
%
%      H = FUERZA_ESTATICO_RETICULADA returns the handle to a new
FUERZA_ESTATICO_RETICULADA or the handle to
%      the existing singleton*.
%
%
FUERZA_ESTATICO_RETICULADA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in FUERZA_ESTATICO_RETICULADA.M with
the given input arguments.
%
%      FUERZA_ESTATICO_RETICULADA('Property','Value',...) creates a
new FUERZA_ESTATICO_RETICULADA or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Fuerza_Estatico_Reticulada_OpeningFcn
gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Fuerza_Estatico_Reticulada_OpeningFcn via varargin.
%
```

```
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Fuerza_Estatico_Reticulada

% Last Modified by GUIDE v2.5 12-Nov-2011 20:31:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Fuerza_Estatico_Reticulada_OpeningFcn, ...
                  'gui_OutputFcn',    @Fuerza_Estatico_Reticulada_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Fuerza_Estatico_Reticulada is made visible.
function Fuerza_Estatico_Reticulada_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Fuerza_Estatico_Reticulada (see
VARARGIN)

% Choose default command line output for Fuerza_Estatico_Reticulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Fuerza_Estatico_Reticulada wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Fuerza_Estatico_Reticulada_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;

global numero_aplicfuerzas DatosFuerzas
%se muestra en pantalla el valor de las variable numero_aplicfuerzas y
%DatosFuerza
set(handles.numero_fuerzas_output,'String',numero_aplicfuerzas)
set(handles.fuerzas_output,'data',DatosFuerzas)

% --- Executes on button press in
numeronodosFuerza_Estatico_Reticulada.
function numeronodosFuerza_Estatico_Reticulada_Callback(hObject,
eventdata, handles)
% hObject    handle to numeronodosFuerza_Estatico_Reticulada (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana Numero_Fuerzas_Estatico_ER
Numero_Fuerzas_Estatico_ER

uiwait

global numero_aplicfuerzas
%se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numero_fuerzas_output,'String',numero_aplicfuerzas)

function numero_fuerzas_output_Callback(hObject, eventdata, handles)
% hObject    handle to numero_fuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numero_fuerzas_output as text
%         str2double(get(hObject,'String')) returns contents of
numero_fuerzas_output as a double

% --- Executes during object creation, after setting all properties.
function numero_fuerzas_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numero_fuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
magnitudfuerzaFuerza_Estatico_Reticulada.
function magnitudfuerzaFuerza_Estatico_Reticulada_Callback(hObject,
eventdata, handles)
% hObject    handle to magnitudfuerzaFuerza_Estatico_Reticulada (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_nodos FuerzaNodos numero_aplicfuerzas contadorfueestre
```

#### DatosFuerzas

```
%se obtiene la matriz FuerzaNodos a partir de los datos introducidos
en la
%ventana Definicion_Fuerzas_Estatico_ER
FuerzaNodos=zeros(numero_nodos,3);
fuercolvector=[1:1:numero_nodos];
fuercol=fuercolvector';

for i=1:numero_aplicfuerzas
    contadorfueestre=i;
Definicion_Fuerzas_Estatico_ER

uiwait

global numeroaplicfuerzas fuerzaaaah fuerzaaav momento
FuerzaNodos(numeroaplicfuerzas,:)=[fuerzaaaah fuerzaaav momento];

end
DatosFuerzas=[fuercol FuerzaNodos];
%Se muestra en en una tabla en pantalla la matriz DatosFuerza que
indica la
%magnitud y sentido de fuerza aplicada en cada nodo
set(handles.fuerzas_output,'data',DatosFuerzas)
% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)

%se cierra la ventana
close Fuerza_Estatico_Reticulada
```

#### Definición de fuerzas estáticas para la estructura articulada

```
function varargout = Fuerza_Estatico_Articulada(varargin)
% FUERZA_ESTATICO_ARTICULADA M-file for Fuerza_Estatico_Articulada.fig
%   FUERZA_ESTATICO_ARTICULADA, by itself, creates a new
FUERZA_ESTATICO_ARTICULADA or raises the existing
%   singleton*.
%
%   H = FUERZA_ESTATICO_ARTICULADA returns the handle to a new
FUERZA_ESTATICO_ARTICULADA or the handle to
%   the existing singleton*.
%
%
FUERZA_ESTATICO_ARTICULADA('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in FUERZA_ESTATICO_ARTICULADA.M with
the given input arguments.
%
%   FUERZA_ESTATICO_ARTICULADA('Property','Value',...) creates a
new FUERZA_ESTATICO_ARTICULADA or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before Fuerza_Estatico_Articulada_OpeningFcn
gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to
```



```
Fuerza_Estatico_Articulada_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Fuerza_Estatico_Articulada

% Last Modified by GUIDE v2.5 12-Nov-2011 20:44:57

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Fuerza_Estatico_Articulada_OpeningFcn, ...
                  'gui_OutputFcn',    @Fuerza_Estatico_Articulada_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Fuerza_Estatico_Articulada is made visible.
function Fuerza_Estatico_Articulada_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Fuerza_Estatico_Articulada (see
VARARGIN)

% Choose default command line output for Fuerza_Estatico_Articulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Fuerza_Estatico_Articulada wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Fuerza_Estatico_Articulada_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_aplicfuerzas DatosFuerzas
%se muestra en pantalla el valor de las variable numero_aplicfuerzas y
%DatosFuerza
set(handles.numfuerzas_output, 'String', numero_aplicfuerzas)
set(handles.uitable2, 'data', DatosFuerzas)

% --- Executes on button press in
numeronodosfuerza_Fuerza_Estatico_Articulada.
function
numeronodosfuerza_Fuerza_Estatico_Articulada_Callback(hObject,
eventdata, handles)
% hObject    handle to numeronodosfuerza_Fuerza_Estatico_Articulada
% (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana Numero_Fuerzas_Estatico_EA
Numero_Fuerzas_Estatico_EA

uiwait

global numero_aplicfuerzas
%se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numfuerzas_output, 'String', numero_aplicfuerzas)

function numfuerzas_output_Callback(hObject, eventdata, handles)
% hObject    handle to numfuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of numfuerzas_output
% as text
%         str2double(get(hObject, 'String')) returns contents of
% numfuerzas_output as a double

% --- Executes during object creation, after setting all properties.
function numfuerzas_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numfuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
% called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in
sentidofuerzas_Fuerza_Estatico_Articulada.
function sentidofuerzas_Fuerza_Estatico_Articulada_Callback(hObject,
eventdata, handles)
% hObject    handle to sentidofuerzas_Fuerza_Estatico_Articulada (see
% GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global numero_nodos FuerzaNodos numero_aplicfuerzas contadorfueestar
DatosFuerzas

%se obtiene la matriz FuerzaNodos a partir de los datos introducidos
en la
%ventana Definicion_Fuerzas_Estatico_EA
FuerzaNodos=zeros(numero_nodos,2);
fuercolvector=[1:1:numero_nodos];
fuercol=fuercolvector';

for i=1:numero_aplicfuerzas
    contadorfueestar=i;
    Definicion_Fuerzas_Estatico_EA

uiwait

global numeroaplicfuerzas fuerzaaah fuerzaaav
FuerzaNodos(numeroaplicfuerzas,:)=[fuerzaaah fuerzaaav];

end
DatosFuerzas=[fuercol FuerzaNodos];
%Se muestra en en una tabla en pantalla la matriz DatosFuerza que
indica la
%magnitud y sentido de fuerza aplicada en cada nodo
set(handles.uitable2,'data',DatosFuerzas)

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)

%se cierra la ventana

close Fuerza_Estatico_Articulada
```

### Número de fuerzas estáticas aplicadas en la estructura reticulada

```
function varargout = Numero_Fuerzas_Estatico_ER(varargin)
% NUMERO_FUERZAS_ESTATICO_ER M-file for Numero_Fuerzas_Estatico_ER.fig
%     NUMERO_FUERZAS_ESTATICO_ER, by itself, creates a new
NUMERO_FUERZAS_ESTATICO_ER or raises the existing
%     singleton*.
%
%     H = NUMERO_FUERZAS_ESTATICO_ER returns the handle to a new
NUMERO_FUERZAS_ESTATICO_ER or the handle to
%     the existing singleton*.
%
%
NUMERO_FUERZAS_ESTATICO_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_FUERZAS_ESTATICO_ER.M with
the given input arguments.
%
%     NUMERO_FUERZAS_ESTATICO_ER('Property','Value',...) creates a
new NUMERO_FUERZAS_ESTATICO_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_Fuerzas_Estatico_ER_OpeningFcn
gets called. An
%     unrecognized property name or invalid value makes property
```

```
application
%      stop. All inputs are passed to
Numero_Fuerzas_Estatico_ER_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Numero_Fuerzas_Estatico_ER

% Last Modified by GUIDE v2.5 19-Nov-2011 20:04:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Numero_Fuerzas_Estatico_ER_OpeningFcn, ...
                  'gui_OutputFcn',   @Numero_Fuerzas_Estatico_ER_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Fuerzas_Estatico_ER is made visible.
function Numero_Fuerzas_Estatico_ER_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Fuerzas_Estatico_ER (see
VARARGIN)

% Choose default command line output for Numero_Fuerzas_Estatico_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Fuerzas_Estatico_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Fuerzas_Estatico_ER_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%se cierra la vetana
close Numero_Fuerzas_Estatico_ER

function numnodos_input_Callback(hObject, eventdata, handles)
% hObject      handle to numnodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodos_input as
text
%          str2double(get(hObject,'String')) returns contents of
numnodos_input as a double

% --- Executes during object creation, after setting all properties.
function numnodos_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numnodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numnodos_output_Callback(hObject, eventdata, handles)
% hObject      handle to numnodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numnodos_output as
text
%          str2double(get(hObject,'String')) returns contents of
numnodos_output as a double

% --- Executes during object creation, after setting all properties.
function numnodos_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numnodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end  
  
% --- Executes on button press in aceptar.  
function aceptar_Callback(hObject, eventdata, handles)  
% hObject    handle to aceptar (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
global numero_aplicfuerzas  
%Se asigna el valor introducido en la edit text a la variable  
%numero_aplicfuerzas  
numero_aplicfuerzas=str2double(get(handles.numnodos_input,'String'));  
%Se muestra en pantalla el valor de la variable numero_aplicfuerzas  
set(handles.numnodos_output,'String',numero_aplicfuerzas);
```

### Número de fuerzas estáticas aplicadas en la estructura articulada

```
function varargout = Numero_Fuerzas_Estatico_EA(varargin)  
% NUMERO_FUERZAS_ESTATICO_EA M-file for Numero_Fuerzas_Estatico_EA.fig  
%     NUMERO_FUERZAS_ESTATICO_EA, by itself, creates a new  
%     NUMERO_FUERZAS_ESTATICO_EA or raises the existing  
%     singleton*.  
%  
%     H = NUMERO_FUERZAS_ESTATICO_EA returns the handle to a new  
%     NUMERO_FUERZAS_ESTATICO_EA or the handle to  
%     the existing singleton*.  
%  
%  
%     NUMERO_FUERZAS_ESTATICO_EA('CALLBACK',hObject,eventData,handles,...)  
%     calls the local  
%     function named CALLBACK in NUMERO_FUERZAS_ESTATICO_EA.M with  
%     the given input arguments.  
%  
%     NUMERO_FUERZAS_ESTATICO_EA('Property','Value',...) creates a  
%     new NUMERO_FUERZAS_ESTATICO_EA or raises the  
%     existing singleton*. Starting from the left, property value  
%     pairs are  
%     applied to the GUI before Numero_Fuerzas_Estatico_EA_OpeningFcn  
%     gets called. An  
%     unrecognized property name or invalid value makes property  
%     application  
%     stop. All inputs are passed to  
%     Numero_Fuerzas_Estatico_EA_OpeningFcn via varargin.  
%  
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
%     only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help  
Numero_Fuerzas_Estatico_EA  
  
% Last Modified by GUIDE v2.5 19-Nov-2011 20:00:56  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',  gui_Singleton, ...
```

```
'gui_OpeningFcn',
@Numero_Fuerzas_Estatico_EA_OpeningFcn, ...
'gui_OutputFcn',
@Numero_Fuerzas_Estatico_EA_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_Fuerzas_Estatico_EA is made visible.
function Numero_Fuerzas_Estatico_EA_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_Fuerzas_Estatico_EA (see
VARARGIN)

% Choose default command line output for Numero_Fuerzas_Estatico_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_Fuerzas_Estatico_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_Fuerzas_Estatico_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject     handle to cerrar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%se cierra la ventana
close Numero_Fuerzas_Estatico_EA

function numero_input_Callback(hObject, eventdata, handles)
% hObject     handle to numero_input (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numero_input as
text
%      str2double(get(hObject,'String')) returns contents of
numero_input as a double

% --- Executes during object creation, after setting all properties.
function numero_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numero_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numero_output_Callback(hObject, eventdata, handles)
% hObject      handle to numero_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numero_output as
text
%      str2double(get(hObject,'String')) returns contents of
numero_output as a double

% --- Executes during object creation, after setting all properties.
function numero_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numero_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_aplicfuerzas
%Se asigna el valor introducido en la edit text a la variable
numero_aplicfuerzas
numero_aplicfuerzas=str2double(get(handles.numero_input,'String'));
%Se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numero_output,'String',numero_aplicfuerzas);
```



## Características de las fuerzas estáticas aplicadas en la estructura reticulada

```
function varargout = Definicion_Fuerzas_Estatico_ER(varargin)
% DEFINICION_FUERZAS_ESTATICO_ER M-file for
Definicion_Fuerzas_Estatico_ER.fig
%     DEFINICION_FUERZAS_ESTATICO_ER, by itself, creates a new
DEFINICION_FUERZAS_ESTATICO_ER or raises the existing
%     singleton*.
%
%     H = DEFINICION_FUERZAS_ESTATICO_ER returns the handle to a new
DEFINICION_FUERZAS_ESTATICO_ER or the handle to
%     the existing singleton*.
%
%
DEFINICION_FUERZAS_ESTATICO_ER('CALLBACK',hObject,eventData,handles,...
.) calls the local
%     function named CALLBACK in DEFINICION_FUERZAS_ESTATICO_ER.M
with the given input arguments.
%
%     DEFINICION_FUERZAS_ESTATICO_ER('Property','Value',...) creates
a new DEFINICION_FUERZAS_ESTATICO_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Definicion_Fuerzas_Estatico_ER_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Definicion_Fuerzas_Estatico_ER_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Definicion_Fuerzas_Estatico_ER

% Last Modified by GUIDE v2.5 21-Nov-2011 11:39:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Definicion_Fuerzas_Estatico_ER_OpeningFcn, ...
                  'gui_OutputFcn',   @Definicion_Fuerzas_Estatico_ER_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before Definicion_Fuerzas_Estatico_ER is made
visible.
function Definicion_Fuerzas_Estatico_ER_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Definicion_Fuerzas_Estatico_ER
(see VARARGIN)

% Choose default command line output for
Definicion_Fuerzas_Estatico_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Definicion_Fuerzas_Estatico_ER wait for user response
(see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Definicion_Fuerzas_Estatico_ER_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorfueestre
%se muestra en pantalla el valor de la variable contadorfueestre
set(handles.contadorfueestre_output, 'String', contadorfueestre)

function amplitudx_input_Callback(hObject, eventdata, handles)
% hObject      handle to amplitudx_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of amplitudx_input as
text
%          str2double(get(hObject, 'String')) returns contents of
amplitudx_input as a double

% --- Executes during object creation, after setting all properties.
function amplitudx_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to amplitudx_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
```

end

```
function amplitud_input_Callback(hObject, eventdata, handles)
% hObject      handle to amplitud_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitud_input as
text
%          str2double(get(hObject,'String')) returns contents of
amplitud_input as a double

% --- Executes during object creation, after setting all properties.
function amplitud_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to amplitud_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global fuerzaaah fuerzaaav numeroaplicfuerzas numnoddisfue momento

%se asigna el valor introducido en cada una de las edit text a las
%variables numeroaplicfuerzas fuerzaaah fuerzaaavnumnoddisfue y
momento
numeroaplicfuerzas=str2double(get(handles.numeronodos_input,'String'))
;
fuerzaaah=str2double(get(handles.amplitudx_input,'String'));
fuerzaaav=str2double(get(handles.amplitudy_input,'String'));
numnoddisfue=str2double(get(handles.numeronodos_input,'String'));
momento=str2double(get(handles.momentoz_input,'String'));

close Definicion_Fuerzas_Estatico_ER
```

```
function numeronodos_input_Callback(hObject, eventdata, handles)
% hObject      handle to numeronodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numeronodos_input
as text
%          str2double(get(hObject,'String')) returns contents of
numeronodos_input as a double
```

```
% --- Executes during object creation, after setting all properties.
function numeronodos_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numeronodos_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function momentoz_input_Callback(hObject, eventdata, handles)
% hObject    handle to momentoz_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of momentoz_input as
text
%       str2double(get(hObject,'String')) returns contents of
momentoz_input as a double

% --- Executes during object creation, after setting all properties.
function momentoz_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to momentoz_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadorfueestre_output_Callback(hObject, eventdata, handles)
% hObject    handle to contadorfueestre_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
contadorfueestre_output as text
%       str2double(get(hObject,'String')) returns contents of
contadorfueestre_output as a double

% --- Executes during object creation, after setting all properties.
function contadorfueestre_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to contadorfueestre_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana
close Definicion_Fuerzas_Estatico_ER
```

### Características de las fuerzas estáticas aplicadas en la estructura articulada

```
function varargout = Definicion_Fuerzas_Estatico_EA(varargin)
% DEFINICION_FUERZAS_ESTATICO_EA M-file for
Definicion_Fuerzas_Estatico_EA.fig
%      DEFINICION_FUERZAS_ESTATICO_EA, by itself, creates a new
DEFINICION_FUERZAS_ESTATICO_EA or raises the existing
%      singleton*.
%
%      H = DEFINICION_FUERZAS_ESTATICO_EA returns the handle to a new
DEFINICION_FUERZAS_ESTATICO_EA or the handle to
%      the existing singleton*.
%
%
DEFINICION_FUERZAS_ESTATICO_EA('CALLBACK',hObject,eventData,handles,...
.) calls the local
%      function named CALLBACK in DEFINICION_FUERZAS_ESTATICO_EA.M
with the given input arguments.
%
%      DEFINICION_FUERZAS_ESTATICO_EA('Property','Value',...) creates
a new DEFINICION_FUERZAS_ESTATICO_EA or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before
Definicion_Fuerzas_Estatico_EA_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Definicion_Fuerzas_Estatico_EA_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Definicion_Fuerzas_Estatico_EA

% Last Modified by GUIDE v2.5 21-Nov-2011 11:39:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
```

```
'gui_OpeningFcn',
@Definicion_Fuerzas_Estatico_EA_OpeningFcn, ...
'gui_OutputFcn',
@Definicion_Fuerzas_Estatico_EA_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Definicion_Fuerzas_Estatico_EA is made
% visible.
function Definicion_Fuerzas_Estatico_EA_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Definicion_Fuerzas_Estatico_EA
% (see VARARGIN)

% Choose default command line output for
Definicion_Fuerzas_Estatico_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Definicion_Fuerzas_Estatico_EA wait for user response
% (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Definicion_Fuerzas_Estatico_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorfueestar
%se muestra en pantalla el valor de la variable contadorfueestar
set(handles.contadorfueestar_output, 'String', contadorfueestar)

function amplitudx_input_Callback(hObject, eventdata, handles)
% hObject      handle to amplitudx_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of amplitudx_input as
```

```
text
%      str2double(get(hObject,'String')) returns contents of
amplitudx_input as a double

% --- Executes during object creation, after setting all properties.
function amplitudx_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitudx_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function amplitudy_input_Callback(hObject, eventdata, handles)
% hObject    handle to amplitudy_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitudy_input as
text
%      str2double(get(hObject,'String')) returns contents of
amplitudy_input as a double

% --- Executes during object creation, after setting all properties.
function amplitudy_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitudy_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global fuerzaaah fuerzaaav numeroaplicfuerzas numnoddisfue

%se asigna el valor introducido en cada una de las edit text a las
%variables numeroaplicfuerzas fuerzaaah y fuerzaaavnumnoddisfue
numeroaplicfuerzas=str2double(get(handles.numeronodos_input,'String'))
;
fuerzaaah=str2double(get(handles.amplitudx_input,'String'));
fuerzaaav=str2double(get(handles.amplitudy_input,'String'));
numnoddisfue=str2double(get(handles.numeronodos_input,'String'));
```

```
close Definicion_Fuerzas_Estatico_EA

function numeronodos_input_Callback(hObject, eventdata, handles)
% hObject      handle to numeronodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numeronodos_input
as text
%          str2double(get(hObject,'String')) returns contents of
numeronodos_input as a double

% --- Executes during object creation, after setting all properties.
function numeronodos_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numeronodos_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadorfueestar_output_Callback(hObject, eventdata, handles)
% hObject      handle to contadorfueestar_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
contadorfueestar_output as text
%          str2double(get(hObject,'String')) returns contents of
contadorfueestar_output as a double

% --- Executes during object creation, after setting all properties.
function contadorfueestar_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to contadorfueestar_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```



```
%se cierra la ventana  
close Definicion_Fuerzas_Estatico_EA
```

### Ventana de fuerzas dinámicas aplicadas a la estructura reticulada

```
function varargout = Fuerza_Dinamica_Reticulada(varargin)  
% FUERZA_DINAMICA_RETICULADA M-file for Fuerza_Dinamica_Reticulada.fig  
%     FUERZA_DINAMICA_RETICULADA, by itself, creates a new  
FUERZA_DINAMICA_RETICULADA or raises the existing  
%     singleton*.  
%  
%     H = FUERZA_DINAMICA_RETICULADA returns the handle to a new  
FUERZA_DINAMICA_RETICULADA or the handle to  
%     the existing singleton*.  
%  
%  
FUERZA_DINAMICA_RETICULADA('CALLBACK',hObject,eventData,handles,...)  
calls the local  
%     function named CALLBACK in FUERZA_DINAMICA_RETICULADA.M with  
the given input arguments.  
%  
%     FUERZA_DINAMICA_RETICULADA('Property','Value',...) creates a  
new FUERZA_DINAMICA_RETICULADA or raises the  
%     existing singleton*. Starting from the left, property value  
pairs are  
%     applied to the GUI before Fuerza_Dinamica_Reticulada_OpeningFcn  
gets called. An  
%     unrecognized property name or invalid value makes property  
application  
%     stop. All inputs are passed to  
Fuerza_Dinamica_Reticulada_OpeningFcn via varargin.  
%  
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help  
Fuerza_Dinamica_Reticulada  
  
% Last Modified by GUIDE v2.5 12-Nov-2011 11:06:22  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',  gui_Singleton, ...  
                  'gui_OpeningFcn', @Fuerza_Dinamica_Reticulada_OpeningFcn, ...  
                  'gui_OutputFcn',  @Fuerza_Dinamica_Reticulada_OutputFcn, ...  
                  'gui_LayoutFcn',  [] , ...  
                  'gui_Callback',   []);  
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end  
  
if nargout  
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});  
else  
    gui_mainfcn(gui_State, varargin{:});  
end
```

```
% End initialization code - DO NOT EDIT

% --- Executes just before Fuerza_Dinamica_Reticulada is made visible.
function Fuerza_Dinamica_Reticulada_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Fuerza_Dinamica_Reticulada (see
VARARGIN)

% Choose default command line output for Fuerza_Dinamica_Reticulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Fuerza_Dinamica_Reticulada wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Fuerza_Dinamica_Reticulada_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global FuerzasDinamicoReticulada numero_aplicfuerzas tini tfin

%Se muestra en pantalla los valores de las variables
%FuerzasDinamicoReticulada numero_aplicfuerzas tini y tfin
set(handles.caractfuerzas_output,'data',FuerzasDinamicoReticulada)
set(handles.numerodenodos_output,'String',numero_aplicfuerzas)
set(handles.tiempoinicial_output,'String',tini)
set(handles.tiempofinal_output,'String',tfin)

% --- Executes on button press in
caractfuerzas_Fuerza_Dinamica_Reticulada.
function caractfuerzas_Fuerza_Dinamica_Reticulada_Callback(hObject,
eventdata, handles)
% hObject    handle to caractfuerzas_Fuerza_Dinamica_Reticulada (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global numero_nodos FuerzaNodos numero_aplicfuerzas omegaaa desfase
contadorfuedinret FuerzasDinamicoReticulada

%se obtienen las matrices FuerzasNodos omegaaa y desfase a partir de
los
%datos introducidos en la ventana Definicion_Fuerzas_ER
FuerzaNodos=zeros(numero_nodos,3);
fuercolvector=[1:1:numero_nodos];
fuercol=fuercolvector';
omegaaa=zeros(numero_nodos,1);
desfase=zeros(numero_nodos,1);
```

```
for i=1:numero_aplicfuerzas
    contadorfuedinret=i;
Definicion_Fuerzas_ER

uiwait

global numeroaplicfuerzas fuerzaaaah fuerzaaav giroz omeGa desFase
numnoddisfue
FuerzaNodos(numeroaplicfuerzas,:)= [fuerzaaaah fuerzaaav giroz];
omegaaa(numnoddisfue,:)=omeGa;
desfase(numnoddisfue,:)=desFase;

end
FuerzasDinamicoReticulada=[fuercol FuerzaNodos omeGaaa desfase];
%se muestra en pantalla el valor de la variable
FuerzasDinamicoReticulada
set(handles.caractfuerzas_output,'data',FuerzasDinamicoReticulada)

% --- Executes on button press in
numeroaplicdefuerza_Fuerza_Dinamica_Reticulada.
function
numeroaplicdefuerza_Fuerza_Dinamica_Reticulada_Callback(hObject,
eventdata, handles)
% hObject    handle to numeroaplicdefuerza_Fuerza_Dinamica_Reticulada
(see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana Numero_de_Fuerzas_ER
Numero_de_Fuerzas_ER

uiwait

global numero_aplicfuerzas
%se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numerodenodos_output,'String',numero_aplicfuerzas)

function numerodenodos_output_Callback(hObject, eventdata, handles)
% hObject    handle to numerodenodos_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numerodenodos_output as text
%         str2double(get(hObject,'String')) returns contents of
numerodenodos_output as a double

% --- Executes during object creation, after setting all properties.
function numerodenodos_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numerodenodos_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
        set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
intervalotiempo_Fuerza_Dinamica_Reticulada.
function intervalotiempo_Fuerza_Dinamica_Reticulada_Callback(hObject, eventdata, handles)
% hObject    handle to intervalotiempo_Fuerza_Dinamica_Reticulada (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana IntervaloTiempor
IntervaloTiempor

uiwait

global tini tfin
%se muestra en pantalla los valores de las variables tini y tfin
set(handles.tiempoinicial_output,'String',tini)
set(handles.tiempofinal_output,'String',tfin)

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se cierra la ventana Fuerza_Dinamica_Reticulada
close Fuerza_Dinamica_Reticulada

function tiempoinicial_output_Callback(hObject, eventdata, handles)
% hObject    handle to tiempoinicial_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
tiempoinicial_output as text
%        str2double(get(hObject,'String')) returns contents of
tiempoinicial_output as a double

% --- Executes during object creation, after setting all properties.
function tiempoinicial_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tiempoinicial_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tiempofinal_output_Callback(hObject, eventdata, handles)
% hObject    handle to tiempofinal_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tiempofinal_output
as text
%      str2double(get(hObject,'String')) returns contents of
tiempofinal_output as a double

% --- Executes during object creation, after setting all properties.
function tiempofinal_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to tiempofinal_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Ventana de fuerzas dinámicas aplicadas a la estructura articulada

```
function varargout = Fuerza_Dinamica_Articulada(varargin)
% FUERZA_DINAMICA_ARTICULADA M-file for Fuerza_Dinamica_Articulada.fig
%      FUERZA_DINAMICA_ARTICULADA, by itself, creates a new
FUERZA_DINAMICA_ARTICULADA or raises the existing
%      singleton*.
%
%      H = FUERZA_DINAMICA_ARTICULADA returns the handle to a new
FUERZA_DINAMICA_ARTICULADA or the handle to
%      the existing singleton*.
%
%
FUERZA_DINAMICA_ARTICULADA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in FUERZA_DINAMICA_ARTICULADA.M with
the given input arguments.
%
%      FUERZA_DINAMICA_ARTICULADA('Property','Value',...) creates a
new FUERZA_DINAMICA_ARTICULADA or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Fuerza_Dinamica_Articulada_OpeningFcn
gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Fuerza_Dinamica_Articulada_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Fuerza_Dinamica_Articulada

% Last Modified by GUIDE v2.5 19-Nov-2011 18:26:53
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Fuerza_Dinamica_Articulada_OpeningFcn, ...
                  'gui_OutputFcn',  @Fuerza_Dinamica_Articulada_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Fuerza_Dinamica_Articulada is made visible.
function Fuerza_Dinamica_Articulada_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Fuerza_Dinamica_Articulada (see
VARARGIN)

% Choose default command line output for Fuerza_Dinamica_Articulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Fuerza_Dinamica_Articulada wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Fuerza_Dinamica_Articulada_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_aplicfuerzas DatosFuerzas tini tfin

%Se muestra en pantalla los valores de las variables
%DatosFuerzas numero_aplicfuerzas tini y tfin
set(handles.numerodefuerzas_output, 'String', numero_aplicfuerzas)
set(handles.tabladatosfuerzas_output, 'data', DatosFuerzas)
set(handles.tfin_output, 'String', tfin)
set(handles.tini_output, 'String', tini)
```

```
% --- Executes on button press in
numeroaplicaciondefuerzas_Fuerza_Dinamica_Articulada.
function
numeroaplicaciondefuerzas_Fuerza_Dinamica_Articulada_Callback(hObject,
eventdata, handles)
% hObject    handle to
numeroaplicaciondefuerzas_Fuerza_Dinamica_Articulada (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana Numero_de_Fuerzas_EA
Numero_de_Fuerzas_EA

uiwait

global numero_aplicfuerzas
%se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numeroedefuerzas_output, 'String', numero_aplicfuerzas)

function numeroedefuerzas_output_Callback(hObject, eventdata, handles)
% hObject    handle to numeroedefuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of
numeroedefuerzas_output as text
%         str2double(get(hObject, 'String')) returns contents of
numeroedefuerzas_output as a double

% --- Executes during object creation, after setting all properties.
function numeroedefuerzas_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numeroedefuerzas_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in
aplicaciondefuerza_Fuerza_Dinamica_Articulada.
function
aplicaciondefuerza_Fuerza_Dinamica_Articulada_Callback(hObject,
eventdata, handles)
% hObject    handle to aplicaciondefuerza_Fuerza_Dinamica_Articulada
(see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_nodos FuerzaNodos numero_aplicfuerzas omegaaa desfase
contadorfuedinart DatosFuerzas

%se obtienen las matrices FuerzasNodos omegaaa y desfase a partir de
los
```

```
%datos introducidos en la ventana Definicion_Fuerzas_EA
FuerzaNodos=zeros(numero_nodos,2);
fuercolvector=[1:1:numero_nodos];
fuercol=fuercolvector';
omegaaa=zeros(numero_nodos,1);
desfase=zeros(numero_nodos,1);

for i=1:numero_aplicfuerzas
    contadorfuedinart=i;
Definicion_Fuerzas_EA

uiwait

global numeroaplicfuerzas fuerzaaaah fuerzaaav omeGa    desFase
numnoddisfue
FuerzaNodos(numeroaplicfuerzas,:)=[fuerzaaaah fuerzaaav];
omegaaa(numnoddisfue,:)=omeGa;
desfase(numnoddisfue,:)=desFase;
DatosFuerzas=[fuercol FuerzaNodos omeaaa desfase];
%se muestra en pantalla el valor de la variable
FuerzasDinamicoReticulada
set(handles.tabladatosfuerzas_output,'data',DatosFuerzas)
end

% --- Executes on button press in
intervalotiempo_Fuerza_Dinamica_Articulada.
function intervalotiempo_Fuerza_Dinamica_Articulada_Callback(hObject,
eventdata, handles)
% hObject    handle to intervalotiempo_Fuerza_Dinamica_Articulada (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se abre la ventana IntervaloTiempo
IntervaloTiempo

uiwait

global tini tfin
%se muestra en pantalla los valores de las variables tini y tfin
set(handles.tfin_output,'String',tfin)
set(handles.tini_output,'String',tini)

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se cierra la ventana Fuerza_Dinamica_Articulada
close Fuerza_Dinamica_Articulada

function tini_output_Callback(hObject, eventdata, handles)
% hObject    handle to tini_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```



```
% Hints: get(hObject,'String') returns contents of tini_output as text
%         str2double(get(hObject,'String')) returns contents of
tini_output as a double

% --- Executes during object creation, after setting all properties.
function tini_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tini_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tfm_output_Callback(hObject, eventdata, handles)
% hObject    handle to tfm_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tfm_output as text
%         str2double(get(hObject,'String')) returns contents of
tfm_output as a double

% --- Executes during object creation, after setting all properties.
function tfm_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tfm_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Número de fuerzas dinámicas aplicadas a la estructura reticulada

```
function varargout = Numero_de_Fuerzas_ER(varargin)
% NUMERO_DE_FUERZAS_ER M-file for Numero_de_Fuerzas_ER.fig
%     NUMERO_DE_FUERZAS_ER, by itself, creates a new
NUMERO_DE_FUERZAS_ER or raises the existing
%     singleton*.
%
%     H = NUMERO_DE_FUERZAS_ER returns the handle to a new
NUMERO_DE_FUERZAS_ER or the handle to
%     the existing singleton*.
%
%     NUMERO_DE_FUERZAS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_DE_FUERZAS_ER.M with the
given input arguments.
```

```
%
%     NUMERO_DE_FUERZAS_ER('Property','Value',...) creates a new
NUMERO_DE_FUERZAS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_de_Fuerzas_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_de_Fuerzas_ER_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Numero_de_Fuerzas_ER

% Last Modified by GUIDE v2.5 19-Nov-2011 18:55:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_de_Fuerzas_ER_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Numero_de_Fuerzas_ER_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Numero_de_Fuerzas_ER is made visible.
function Numero_de_Fuerzas_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_de_Fuerzas_ER (see
VARARGIN)

% Choose default command line output for Numero_de_Fuerzas_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_de_Fuerzas_ER wait for user response (see
UIRESUME)
```

```
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Numero_de_Fuerzas_ER_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana Numero_de_Fuerzas_ER
close Numero_de_Fuerzas_ER

function numaplicfuerz_input_Callback(hObject, eventdata, handles)
% hObject      handle to numaplicfuerz_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numaplicfuerz_input
as text
%           str2double(get(hObject,'String')) returns contents of
numaplicfuerz_input as a double

% --- Executes during object creation, after setting all properties.
function numaplicfuerz_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to numaplicfuerz_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function numaplicfuerz_output_Callback(hObject, eventdata, handles)
% hObject      handle to numaplicfuerz_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numaplicfuerz_output as text
%           str2double(get(hObject,'String')) returns contents of
numaplicfuerz_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function numaplicfuerz_output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numaplicfuerz_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global numero_aplicfuerzas
%Se asigna el valor introducido en la edit text a la variable
%numero_aplicfuerzas
numero_aplicfuerzas=str2double(get(handles.numaplicfuerz_input,'String
'));
%Se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numaplicfuerz_output,'String',numero_aplicfuerzas);
```

### Número de fuerzas dinámicas aplicadas a la estructura articulada

```
function varargout = Numero_de_Fuerzas_EA(varargin)
% NUMERO_DE_FUERZAS_EA M-file for Numero_de_Fuerzas_EA.fig
%     NUMERO_DE_FUERZAS_EA, by itself, creates a new
NUMERO_DE_FUERZAS_EA or raises the existing
%     singleton*.
%
%     H = NUMERO_DE_FUERZAS_EA returns the handle to a new
NUMERO_DE_FUERZAS_EA or the handle to
%     the existing singleton*.
%
%     NUMERO_DE_FUERZAS_EA('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in NUMERO_DE_FUERZAS_EA.M with the
given input arguments.
%
%     NUMERO_DE_FUERZAS_EA('Property','Value',...) creates a new
NUMERO_DE_FUERZAS_EA or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Numero_de_Fuerzas_EA_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Numero_de_Fuerzas_EA_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help
Numero_de_Fuerzas_EA

% Last Modified by GUIDE v2.5 19-Nov-2011 18:44:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Numero_de_Fuerzas_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Numero_de_Fuerzas_EA_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Numero_de_Fuerzas_EA is made visible.
function Numero_de_Fuerzas_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Numero_de_Fuerzas_EA (see
VARARGIN)

% Choose default command line output for Numero_de_Fuerzas_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Numero_de_Fuerzas_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Numero_de_Fuerzas_EA_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Se cierra la ventana Numero_de_Fuerzas_EA
close Numero_de_Fuerzas_EA

function numaplicfuerz_input_Callback(hObject, eventdata, handles)
% hObject handle to numaplicfuerz_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numaplicfuerz_input
as text
% str2double(get(hObject,'String')) returns contents of
numaplicfuerz_input as a double

% --- Executes during object creation, after setting all properties.
function numaplicfuerz_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to numaplicfuerz_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function numaplicfuerz_output_Callback(hObject, eventdata, handles)
% hObject handle to numaplicfuerz_output (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
numaplicfuerz_output as text
% str2double(get(hObject,'String')) returns contents of
numaplicfuerz_output as a double

% --- Executes during object creation, after setting all properties.
function numaplicfuerz_output_CreateFcn(hObject, eventdata, handles)
% hObject handle to numaplicfuerz_output (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject handle to aceptar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)
global numero_aplicfuerzas
%Se asigna el valor introducido en la edit text a la variable
%numero_aplicfuerzas
numero_aplicfuerzas=str2double(get(handles.numaplicfuerz_input,'String'
'));
%Se muestra en pantalla el valor de la variable numero_aplicfuerzas
set(handles.numaplicfuerz_output,'String',numero_aplicfuerzas);
```

### Definición de fuerzas dinámicas aplicadas a la estructura reticulada

```
function varargout = Definicion_Fuerzas_ER(varargin)
% DEFINICION_FUERZAS_ER M-file for Definicion_Fuerzas_ER.fig
%     DEFINICION_FUERZAS_ER, by itself, creates a new
DEFINICION_FUERZAS_ER or raises the existing
%     singleton*.
%
%     H = DEFINICION_FUERZAS_ER returns the handle to a new
DEFINICION_FUERZAS_ER or the handle to
%     the existing singleton*.
%
%     DEFINICION_FUERZAS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in DEFINICION_FUERZAS_ER.M with the
given input arguments.
%
%     DEFINICION_FUERZAS_ER('Property','Value',...) creates a new
DEFINICION_FUERZAS_ER or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Definicion_Fuerzas_ER_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Definicion_Fuerzas_ER_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Definicion_Fuerzas_ER

% Last Modified by GUIDE v2.5 20-Nov-2011 19:53:50

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Definicion_Fuerzas_ER_OpeningFcn, ...
                  'gui_OutputFcn',   @Definicion_Fuerzas_ER_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Definicion_Fuerzas_ER is made visible.
function Definicion_Fuerzas_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Definicion_Fuerzas_ER (see
VARARGIN)

% Choose default command line output for Definicion_Fuerzas_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Definicion_Fuerzas_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Definicion_Fuerzas_ER_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorfuedinret
%Se muestra en pantalla el valor de la variable contadorfuedinret
set(handles.contadorfuedinret_output, 'String', contadorfuedinret)

function fuerzax_input_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzax_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of fuerzax_input as
text
%          str2double(get(hObject, 'String')) returns contents of
fuerzax_input as a double

% --- Executes during object creation, after setting all properties.
function fuerzax_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to fuerzax_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```



```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fuerzay_input_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzay_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fuerzay_input as
text
%      str2double(get(hObject,'String')) returns contents of
fuerzay_input as a double

% --- Executes during object creation, after setting all properties.
function fuerzay_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to fuerzay_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global fuerzaaah fuerzaaav giroz numeroaplicfuerzas numnoddisfue
desFase omeGa

%Se asigna el valor introducido en cada una de las edit text a las
%variables numeomeroaplicfuerzas giroz fuerzaaah fuerzaaav
numnoddisfue omeGa
%y desFase
numeroaplicfuerzas=str2double(get(handles.edit3,'String'));
fuerzaaah=str2double(get(handles.fuerzax_input,'String'));
fuerzaaav=str2double(get(handles.fuerzay_input,'String'));
giroz=str2double(get(handles.fuerzaz_input,'String'));
numnoddisfue=str2double(get(handles.edit3,'String'));

omeGa=str2double(get(handles.omeg_input,'String'));
desFase=str2double(get(handles.desfas_input,'String'));

%Se cierra la ventana Definicion_Fuerzas_ER
close Definicion_Fuerzas_ER

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3
as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fuerzaz_input_Callback(hObject, eventdata, handles)
% hObject    handle to fuerzaz_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fuerzaz_input as
text
%         str2double(get(hObject,'String')) returns contents of
fuerzaz_input as a double

% --- Executes during object creation, after setting all properties.
function fuerzaz_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fuerzaz_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function omeg_input_Callback(hObject, eventdata, handles)
% hObject    handle to omeg_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of omeg_input as text
%         str2double(get(hObject,'String')) returns contents of
omeg_input as a double

% --- Executes during object creation, after setting all properties.
function omeg_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to omeg_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles      empty - handles not created until after all CreateFcns  
called
```

```
% Hint: edit controls usually have a white background on Windows.  
%      See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function desfas_input_Callback(hObject, eventdata, handles)  
% hObject      handle to desfas_input (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of desfas_input as  
text  
%      str2double(get(hObject,'String')) returns contents of  
desfas_input as a double
```

```
% --- Executes during object creation, after setting all properties.  
function desfas_input_CreateFcn(hObject, eventdata, handles)  
% hObject      handle to desfas_input (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      empty - handles not created until after all CreateFcns  
called
```

```
% Hint: edit controls usually have a white background on Windows.  
%      See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function contadorfuedinret_output_Callback(hObject, eventdata,  
handles)  
% hObject      handle to contadorfuedinret_output (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of  
contadorfuedinret_output as text  
%      str2double(get(hObject,'String')) returns contents of  
contadorfuedinret_output as a double
```

```
% --- Executes during object creation, after setting all properties.  
function contadorfuedinret_output_CreateFcn(hObject, eventdata,  
handles)  
% hObject      handle to contadorfuedinret_output (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      empty - handles not created until after all CreateFcns  
called
```

```
% Hint: edit controls usually have a white background on Windows.  
%      See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))
```

```
        set(hObject, 'BackgroundColor', 'white');  
end
```

### Definición de fuerzas dinámicas aplicadas a la estructura articulada

```
function varargout = Definicion_Fuerzas_EA(varargin)  
% DEFINICION_FUERZAS_EA M-file for Definicion_Fuerzas_EA.fig  
%     DEFINICION_FUERZAS_EA, by itself, creates a new  
DEFINICION_FUERZAS_EA or raises the existing  
%     singleton*.  
%  
%     H = DEFINICION_FUERZAS_EA returns the handle to a new  
DEFINICION_FUERZAS_EA or the handle to  
%     the existing singleton*.  
%  
%     DEFINICION_FUERZAS_EA('CALLBACK',hObject,eventData,handles,...)  
calls the local  
%     function named CALLBACK in DEFINICION_FUERZAS_EA.M with the  
given input arguments.  
%  
%     DEFINICION_FUERZAS_EA('Property','Value',...) creates a new  
DEFINICION_FUERZAS_EA or raises the  
%     existing singleton*. Starting from the left, property value  
pairs are  
%     applied to the GUI before Definicion_Fuerzas_EA_OpeningFcn gets  
called. An  
%     unrecognized property name or invalid value makes property  
application  
%     stop. All inputs are passed to  
Definicion_Fuerzas_EA_OpeningFcn via varargin.  
%  
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help  
Definicion_Fuerzas_EA  
  
% Last Modified by GUIDE v2.5 20-Nov-2011 19:49:41  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',  gui_Singleton, ...  
                  'gui_OpeningFcn', @Definicion_Fuerzas_EA_OpeningFcn, ...  
                  'gui_OutputFcn',  @Definicion_Fuerzas_EA_OutputFcn, ...  
                  'gui_LayoutFcn',  [] , ...  
                  'gui_Callback',   []);  
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end  
  
if nargout  
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});  
else  
    gui_mainfcn(gui_State, varargin{:});  
end  
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before Definicion_Fuerzas_EA is made visible.
function Definicion_Fuerzas_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Definicion_Fuerzas_EA (see
VARARGIN)

% Choose default command line output for Definicion_Fuerzas_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Definicion_Fuerzas_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Definicion_Fuerzas_EA_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global contadorfuedinart

%se muestra en pantalla el valor de la variable contadorfuedinart
set(handles.contadorfuedinart_output, 'String', contadorfuedinart)

function fuerzax_input_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzax_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of fuerzax_input as
text
%           str2double(get(hObject, 'String')) returns contents of
fuerzax_input as a double

% --- Executes during object creation, after setting all properties.
function fuerzax_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to fuerzax_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function fuerzay_input_Callback(hObject, eventdata, handles)
% hObject      handle to fuerzay_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fuerzay_input as
text
%          str2double(get(hObject,'String')) returns contents of
fuerzay_input as a double

% --- Executes during object creation, after setting all properties.
function fuerzay_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to fuerzay_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global fuerzaaah fuerzaaav numeroaplicfuerzas omeGa numnoddisfue
desFase

%Se asigna el valor introducido en cada una de las edit text a las
%variables numeomeroaplicfuerzas fuerzaaah fuerzaaav numnoddisfue
omeGa
%y desFase
numeroaplicfuerzas=str2double(get(handles.edit3,'String'));
fuerzaaah=str2double(get(handles.fuerzax_input,'String'));
fuerzaaav=str2double(get(handles.fuerzay_input,'String'));
numnoddisfue=str2double(get(handles.edit3,'String'));

omeGa=str2double(get(handles.omeg_input,'String'));
desFase=str2double(get(handles.desfas_input,'String'));

%Se cierra la ventana Definicion_Fuerzas_EA
close Definicion_Fuerzas_EA

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%          str2double(get(hObject,'String')) returns contents of edit3
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function omeg_input_Callback(hObject, eventdata, handles)
% hObject    handle to omeg_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of omeg_input as text
%         str2double(get(hObject,'String')) returns contents of
omeg_input as a double

% --- Executes during object creation, after setting all properties.
function omeg_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to omeg_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function desfas_input_Callback(hObject, eventdata, handles)
% hObject    handle to desfas_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of desfas_input as
text
%         str2double(get(hObject,'String')) returns contents of
desfas_input as a double

% --- Executes during object creation, after setting all properties.
function desfas_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to desfas_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function contadorfuedinart_output_Callback(hObject, eventdata,
handles)
% hObject      handle to contadorfuedinart_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
contadorfuedinart_output as text
%          str2double(get(hObject,'String')) returns contents of
contadorfuedinart_output as a double

% --- Executes during object creation, after setting all properties.
function contadorfuedinart_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to contadorfuedinart_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

## Intervalo de tiempo en la estructura reticulada

```
function varargout = IntervaloTiempor(varargin)
% INTERVALOTIEMPOR M-file for IntervaloTiempor.fig
%          INTERVALOTIEMPOR, by itself, creates a new INTERVALOTIEMPOR or
raises the existing
%          singleton*.
%
%          H = INTERVALOTIEMPOR returns the handle to a new
INTERVALOTIEMPOR or the handle to
%          the existing singleton*.
%
%          INTERVALOTIEMPOR('CALLBACK',hObject,eventData,handles,...)
calls the local
%          function named CALLBACK in INTERVALOTIEMPOR.M with the given
input arguments.
%
%          INTERVALOTIEMPOR('Property','Value',...) creates a new
INTERVALOTIEMPOR or raises the
%          existing singleton*. Starting from the left, property value
pairs are
%          applied to the GUI before IntervaloTiempor_OpeningFcn gets
called. An
%          unrecognized property name or invalid value makes property
application
%          stop. All inputs are passed to IntervaloTiempor_OpeningFcn via
varargin.
%
%          *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
```



```
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help IntervaloTiempor

% Last Modified by GUIDE v2.5 03-Nov-2011 15:19:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @IntervaloTiempor_OpeningFcn, ...
                  'gui_OutputFcn',  @IntervaloTiempor_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before IntervaloTiempor is made visible.
function IntervaloTiempor_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to IntervaloTiempor (see VARARGIN)

% Choose default command line output for IntervaloTiempor
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes IntervaloTiempor wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = IntervaloTiempor_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function tiemfin_input_Callback(hObject, eventdata, handles)
% hObject    handle to tiemfin_input (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tiemfin_input as
text
% str2double(get(hObject,'String')) returns contents of
tiemfin_input as a double

% --- Executes during object creation, after setting all properties.
function tiemfin_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to tiemfin_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject handle to aceptar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global tini tfin
%Se asigna el valor introducido en cada una de las edit text a las
%variables tini y tfin
tini=str2double(get(handles.tiemini_input,'String'));
tfin=str2double(get(handles.tiemfin_input,'String'));
%Se cierra la ventana IntervaloTiempor

close IntervaloTiempor

function tiemini_input_Callback(hObject, eventdata, handles)
% hObject handle to tiemini_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tiemini_input as
text
% str2double(get(hObject,'String')) returns contents of
tiemini_input as a double

% --- Executes during object creation, after setting all properties.
function tiemini_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to tiemini_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
```

end

## Intervalo de tiempo en la estructura articulada

```
function varargout = IntervaloTiempo(varargin)
% INTERVALOTIEMPO M-file for IntervaloTiempo.fig
%   INTERVALOTIEMPO, by itself, creates a new INTERVALOTIEMPO or
raises the existing
%   singleton*.
%
%   H = INTERVALOTIEMPO returns the handle to a new INTERVALOTIEMPO
or the handle to
%   the existing singleton*.
%
%   INTERVALOTIEMPO('CALLBACK',hObject,eventData,handles,...) calls
the local
%   function named CALLBACK in INTERVALOTIEMPO.M with the given
input arguments.
%
%   INTERVALOTIEMPO('Property','Value',...) creates a new
INTERVALOTIEMPO or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before IntervaloTiempo_OpeningFcn gets
called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to IntervaloTiempo_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help IntervaloTiempo

% Last Modified by GUIDE v2.5 03-Nov-2011 00:53:08

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @IntervaloTiempo_OpeningFcn, ...
                  'gui_OutputFcn',  @IntervaloTiempo_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before IntervaloTiempo is made visible.
function IntervaloTiempo_OpeningFcn(hObject, eventdata, handles,
```

```
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to IntervaloTiempo (see VARARGIN)

% Choose default command line output for IntervaloTiempo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes IntervaloTiempo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = IntervaloTiempo_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function tiemfin_input_Callback(hObject, eventdata, handles)
% hObject    handle to tiemfin_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tiemfin_input as text
%        str2double(get(hObject,'String')) returns contents of tiemfin_input as a double

% --- Executes during object creation, after setting all properties.
function tiemfin_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tiemfin_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global tini tfin
```

```
%Se asigna el valor introducido en cada una de las edit text a las
%variables tini y tfin
tini=str2double(get(handles.tiemini_input,'String'));
tfm=str2double(get(handles.tiemfin_input,'String'));
%Se cierra la ventana IntervaloTiempo
close IntervaloTiempo

function tiemini_input_Callback(hObject, eventdata, handles)
% hObject      handle to tiemini_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tiemini_input as
text
%      str2double(get(hObject,'String')) returns contents of
tiemini_input as a double

% --- Executes during object creation, after setting all properties.
function tiemini_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to tiemini_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### Ventana de resultados de valores y modos propios de la estructura reticulada

```
function varargout = Valores_Propios_ER(varargin)
% VALORES_PROPIOS_ER M-file for Valores_Propios_ER.fig
%      VALORES_PROPIOS_ER, by itself, creates a new VALORES_PROPIOS_ER
or raises the existing
%      singleton*.
%
%      H = VALORES_PROPIOS_ER returns the handle to a new
VALORES_PROPIOS_ER or the handle to
%      the existing singleton*.
%
%      VALORES_PROPIOS_ER('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in VALORES_PROPIOS_ER.M with the given
input arguments.
%
%      VALORES_PROPIOS_ER('Property','Value',...) creates a new
VALORES_PROPIOS_ER or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before Valores_Propios_ER_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to Valores_Propios_ER_OpeningFcn
via varargin.
%
```

```
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Valores_Propios_ER

% Last Modified by GUIDE v2.5 12-Nov-2011 14:02:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Valores_Propios_ER_OpeningFcn,
                  ...
                  'gui_OutputFcn',   @Valores_Propios_ER_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Valores_Propios_ER is made visible.
function Valores_Propios_ER_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valores_Propios_ER (see
VARARGIN)

% Choose default command line output for Valores_Propios_ER
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Valores_Propios_ER wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Valores_Propios_ER_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;

global ValoresPropiosReticulada numero_nodos RestriccionesNodo

contador=0;
for i=1:numero_nodos
    for I=1:3
        if RestriccionesNodo(i,I)==0
            contador=contador;
        elseif (RestriccionesNodo(i,I)==1) && (I==1)
            contador=contador+1;
            cnames(1,contador)=1+(10*i);

            elseif (RestriccionesNodo(i,I)==1) && (I==2)
                contador=contador+1;
                cnames(1,contador)=2+(10*i);
            elseif (RestriccionesNodo(i,I)==1) && (I==3)
                contador=contador+1;
                cnames(1,contador)=3+(10*i);
            end
        end
    end
end

cnames={0 cnames};

set(handles.uitable2,'data',ValoresPropiosReticulada)
set(handles.uitable2,'ColumnName',cnames)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close Valores_Propios_ER
```

### Ventana de resultados de valores y modos propios de la estructura articulada

```
function varargout = Valores_Propios_EA(varargin)
% VALORES_PROPIOS_EA M-file for Valores_Propios_EA.fig
%   VALORES_PROPIOS_EA, by itself, creates a new VALORES_PROPIOS_EA
%   or raises the existing
%   singleton*.
%
%   H = VALORES_PROPIOS_EA returns the handle to a new
%   VALORES_PROPIOS_EA or the handle to
%   the existing singleton*.
%
%   VALORES_PROPIOS_EA('CALLBACK',hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in VALORES_PROPIOS_EA.M with the given
%   input arguments.
%
%   VALORES_PROPIOS_EA('Property','Value',...) creates a new
%   VALORES_PROPIOS_EA or raises the
%   existing singleton*. Starting from the left, property value
%   pairs are
%   applied to the GUI before Valores_Propios_EA_OpeningFcn gets
%   called. An
```

```
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to Valores_Propios_EA_OpeningFcn
via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Valores_Propios_EA

% Last Modified by GUIDE v2.5 12-Nov-2011 14:02:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Valores_Propios_EA_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @Valores_Propios_EA_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Valores_Propios_EA is made visible.
function Valores_Propios_EA_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Valores_Propios_EA (see
VARARGIN)

% Choose default command line output for Valores_Propios_EA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Valores_Propios_EA wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Valores_Propios_EA_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
```



```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global ValorespropiosArticulada numero_nodos RestriccionesNodo

contador=0;
for i=1:numero_nodos
    for I=1:2
        if RestriccionesNodo(i,I)==0
            contador=contador;
        elseif (RestriccionesNodo(i,I)==1) && (I==1)
            contador=contador+1;
            cnames(1,contador)=1+(10*i);

        elseif (RestriccionesNodo(i,I)==1) && (I==2)
            contador=contador+1;
            cnames(1,contador)=2+(10*i);

        end
    end
end

cnames={0 cnames};

set(handles.uitable2,'data',ValorespropiosArticulada)
set(handles.uitable2,'ColumnName',cnames)

%numeronodos=[1:1:numero_nodos];
%columnaconnumdenodos=numeronodos';

%tablavalorespropios=[columnaconnumdenodos FuerzaNodos
ValorespropiosArticulada desfase];

%set(handles.uitable1,'data',tablavalorespropios)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
close Valores_Propios_EA
```

## Resultados de los desplazamientos en estatico de las estructuras reticuladas

```
function varargout = Calculo_Desplazamientos_ER_Estatico(varargin)
% CALCULO_DESPLAZAMIENTOS_ER_ESTATICO M-file for
% Calculo_Desplazamientos_ER_Estatico.fig
%     CALCULO_DESPLAZAMIENTOS_ER_ESTATICO, by itself, creates a new
%     CALCULO_DESPLAZAMIENTOS_ER_ESTATICO or raises the existing
%     singleton*.
%
%     H = CALCULO_DESPLAZAMIENTOS_ER_ESTATICO returns the handle to a
%     new CALCULO_DESPLAZAMIENTOS_ER_ESTATICO or the handle to
```

```
% the existing singleton*.
%
%
CALCULO_DESPLAZAMIENTOS_ER_ESTATICO('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in
CALCULO_DESPLAZAMIENTOS_ER_ESTATICO.M with the given input arguments.
%
% CALCULO_DESPLAZAMIENTOS_ER_ESTATICO('Property','Value',...)
creates a new CALCULO_DESPLAZAMIENTOS_ER_ESTATICO or raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before
Calculo_Desplazamientos_ER_Estatico_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to
Calculo_Desplazamientos_ER_Estatico_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Calculo_Desplazamientos_ER_Estatico

% Last Modified by GUIDE v2.5 20-Nov-2011 11:26:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn',
@Calculo_Desplazamientos_ER_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',
@Calculo_Desplazamientos_ER_Estatico_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Calculo_Desplazamientos_ER_Estatico is made
visible.
function Calculo_Desplazamientos_ER_Estatico_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to
Calculo_Desplazamientos_ER_Estatico (see VARARGIN)
```

```
% Choose default command line output for
Calculo_Desplazamientos_ER_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Calculo_Desplazamientos_ER_Estatico wait for user
response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout =
Calculo_Desplazamientos_ER_Estatico_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in
caldesp_Calculo_Desplazamientos_ER_Estatico.
function caldesp_Calculo_Desplazamientos_ER_Estatico_Callback(hObject,
eventdata, handles)
% hObject     handle to caldesp_Calculo_Desplazamientos_ER_Estatico
% (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject     handle to cerrar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se cierra la ventana Calculo_Desplazamientos_ER_Estatico
close Calculo_Desplazamientos_ER_Estatico

% --- Executes on button press in
resultados__Calculo_Desplazamientos_ER_Estatico.
function
resultados__Calculo_Desplazamientos_ER_Estatico_Callback(hObject,
eventdata, handles)
% hObject     handle to resultados__Calculo_Desplazamientos_ER_Estatico
% (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global E A I BARRAS Nodo RestriccionesNodo FuerzaNodos Barramuelle
DesplazamientosEstaticoER ans
%Se ejecuta la funcion Desplazamientosenestaticoestructurareticulada y
se
%obtienen los resultados de los desplazamientos gracias a los datos que
hay
%entre parentesis
Desplazamientosenestaticoestructurareticulada(E,A,I,BARRAS,Nodo,Restri
ccionesNodo,FuerzaNodos,Barramuelle);

%Se asignan los resultados obtenidos a la variable
```

```
%DesplazamientosEstaticoER
DesplazamientosEstaticoER=ans;

%Se abre la ventana RsultadosDesp_ER_Estatico
RsultadosDesp_ER_Estatico
%Se mantiene en espera hasta que se cierre la vetana anterior
uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se eliminan todos los datos introducidos y calculados en la interfaz
Y
%con clc se limpia la ventana de comandos
clear all
clc
```

### Resultados de los desplazamientos en estatico de las estructuras articuladas

```
function varargout = Calculo_Desplazamientos_EA_Estatico(varargin)
% CALCULO_DESPLAZAMIENTOS_EA_ESTATICO M-file for
Calculo_Desplazamientos_EA_Estatico.fig
%     CALCULO_DESPLAZAMIENTOS_EA_ESTATICO, by itself, creates a new
CALCULO_DESPLAZAMIENTOS_EA_ESTATICO or raises the existing
%     singleton*.
%
%     H = CALCULO_DESPLAZAMIENTOS_EA_ESTATICO returns the handle to a
new CALCULO_DESPLAZAMIENTOS_EA_ESTATICO or the handle to
%     the existing singleton*.
%
%
CALCULO_DESPLAZAMIENTOS_EA_ESTATICO('CALLBACK',hObject,eventData,handl
es,...) calls the local
%     function named CALLBACK in
CALCULO_DESPLAZAMIENTOS_EA_ESTATICO.M with the given input arguments.
%
%     CALCULO_DESPLAZAMIENTOS_EA_ESTATICO('Property','Value',...)
creates a new CALCULO_DESPLAZAMIENTOS_EA_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Calculo_Desplazamientos_EA_Estatico_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Calculo_Desplazamientos_EA_Estatico_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Calculo_Desplazamientos_EA_Estatico

% Last Modified by GUIDE v2.5 20-Nov-2011 11:24:10

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Calculo_Desplazamientos_EA_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',    @Calculo_Desplazamientos_EA_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Calculo_Desplazamientos_EA_Estatico is made
% visible.
function Calculo_Desplazamientos_EA_Estatico_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to
Calculo_Desplazamientos_EA_Estatico (see VARARGIN)

% Choose default command line output for
Calculo_Desplazamientos_EA_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Calculo_Desplazamientos_EA_Estatico wait for user
% response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout =
Calculo_Desplazamientos_EA_Estatico_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in
caldesp_Calculo_Desplazamientos_EA_Estatico.
function caldesp_Calculo_Desplazamientos_EA_Estatico_Callback(hObject,
eventdata, handles)
% hObject    handle to caldesp_Calculo_Desplazamientos_EA_Estatico
% (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject handle to cerrar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Se cierra la vetana Calculo_Desplazamientos_EA_Estatico
close Calculo_Desplazamientos_EA_Estatico

% --- Executes on button press in
resultados__Calculo_Desplazamientos_EA_Estatico.
function
resultados__Calculo_Desplazamientos_EA_Estatico_Callback(hObject,
eventdata, handles)
% hObject handle to resultados__Calculo_Desplazamientos_EA_Estatico
(see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global E A BARRAS Nodo RestriccionesNodo FuerzaNodos Barramuelle
DesplazamientosEstaticoEA ans
%Se ejecuta la funcion Desplazamientosenestaticoestructuraarticulada y
se
%obtienen los resutados de los desplazamientos gracias a los datos que
hay
%entre parentesis
Desplazamientosenestaticoestructuraarticulada(E,A,BARRAS,Nodo,Restricci
onesNodo,FuerzaNodos,Barramuelle);

%Se asignan los resultados obtenidos a la variable
%DesplazamientosEstaticoEA
DesplazamientosEstaticoEA=ans;

%Se abre la ventana RsultadosDesp_EA_Estatico
RsultadosDesp_EA_Estatico

%Se mantiene en espera hasta que se cierre la vetana anterior
uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject handle to eliminar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Se eliminan todos los datos introducidos y calculados en la interfaz
y
%con clc se limpia la ventana de comandos

clear all
clc
```

### Resultados numéricos de los desplazamientos en estatico de estructuras reticuladas

```
function varargout = RsultadosDesp_ER_Estatico(varargin)
% RESULTADOSDESP_ER_ESTATICO M-file for RsultadosDesp_ER_Estatico.fig
% RESULTADOSDESP_ER_ESTATICO, by itself, creates a new
% RESULTADOSDESP_ER_ESTATICO or raises the existing
% singleton*.
```

```
%
%      H = RESULTADOSDESP_ER_ESTATICO returns the handle to a new
RESULTADOSDESP_ER_ESTATICO or the handle to
%      the existing singleton*.
%
%
RESULTADOSDESP_ER_ESTATICO('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in RESULTADOSDESP_ER_ESTATICO.M with the
given input arguments.
%
%      RESULTADOSDESP_ER_ESTATICO('Property','Value',...) creates a new
RESULTADOSDESP_ER_ESTATICO or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before RsltadosDesp_ER_Estatico_OpeningFcn
gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
RsltadosDesp_ER_Estatico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
RsltadosDesp_ER_Estatico

% Last Modified by GUIDE v2.5 14-Nov-2011 20:06:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @RsltadosDesp_ER_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',   @RsltadosDesp_ER_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before RsltadosDesp_ER_Estatico is made visible.
function RsltadosDesp_ER_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to RsltadosDesp_ER_Estatico (see
```

```
VARARGIN)

% Choose default command line output for RsultadosDesp_ER_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes RsultadosDesp_ER_Estatico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = RsultadosDesp_ER_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global DesplazamientosEstaticoER numero_nodos RestriccionesNodo

%Se crea este bucle para recoger en el vector matrizindicativaR unos
%valores que informaran al usuario que nodo y que desplazamiento es el
que
%se produce inmediatamente a la derecha de su valor
contadorEAed=0; %contador estructura articulada estatico
desplazamientos
for i=1:numero_nodos
    for I=1:3
        if (RestriccionesNodo(i,I)==1) && (I==1) && (contadorEAed==0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR=(10*i)+1;
        elseif (RestriccionesNodo(i,I)==1) && (I==2) &&
(contadorEAed==0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR=(10*i)+2;
        elseif (RestriccionesNodo(i,I)==1) && (I==3) &&
(contadorEAed==0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR=(10*i)+3;
        elseif (RestriccionesNodo(i,I)==1) && (I==1) &&
(contadorEAed~=0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR(contadorEAed,:)=(10*i)+1;
        elseif (RestriccionesNodo(i,I)==1) && (I==2) &&
(contadorEAed~=0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR(contadorEAed,:)=(10*i)+2;
        elseif (RestriccionesNodo(i,I)==1) && (I==3) &&
(contadorEAed~=0)
            contadorEAed=contadorEAed+1;
            matrizindicativaR(contadorEAed,:)=(10*i)+3;
        elseif (RestriccionesNodo(i,I)==0)
            contadorEAed=contadorEAed;
    end
end
```



end

```
DesplazamientosEstaticoERtable=[matrizindicativaR  
DesplazamientosEstaticoER];  
%se muestra una tabla con los valres de los desplazamientos de los  
nodos  
set(handles.uitable1,'data',DesplazamientosEstaticoERtable)
```

```
function desplazamientosERe_output_Callback(hObject, eventdata,  
handles)  
% hObject    handle to desplazamientosERe_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of  
desplazamientosERe_output as text  
%         str2double(get(hObject,'String')) returns contents of  
desplazamientosERe_output as a double
```

```
% --- Executes during object creation, after setting all properties.  
function desplazamientosERe_output_CreateFcn(hObject, eventdata,  
handles)  
% hObject    handle to desplazamientosERe_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function nododesplazazador_output_Callback(hObject, eventdata,  
handles)  
% hObject    handle to nododesplazazador_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of  
nododesplazazador_output as text  
%         str2double(get(hObject,'String')) returns contents of  
nododesplazazador_output as a double
```

```
% --- Executes during object creation, after setting all properties.  
function nododesplazazador_output_CreateFcn(hObject, eventdata,  
handles)  
% hObject    handle to nododesplazazador_output (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns  
called  
  
% Hint: edit controls usually have a white background on Windows.  
%       See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se cierra la ventana ResultadosDesp_ER_Estatico
close ResultadosDesp_ER_Estatico
```

## Resultados numéricos de los desplazamientos en estatico de estructuras articuladas

```
function varargout = ResultadosDesp_EA_Estatico(varargin)
% RESULTADOSDESP_EA_ESTATICO M-file for ResultadosDesp_EA_Estatico.fig
%     RESULTADOSDESP_EA_ESTATICO, by itself, creates a new
%     RESULTADOSDESP_EA_ESTATICO or raises the existing
%     singleton*.
%
%     H = RESULTADOSDESP_EA_ESTATICO returns the handle to a new
%     RESULTADOSDESP_EA_ESTATICO or the handle to
%     the existing singleton*.
%
%
%     RESULTADOSDESP_EA_ESTATICO('CALLBACK',hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in RESULTADOSDESP_EA_ESTATICO.M with the
%     given input arguments.
%
%     RESULTADOSDESP_EA_ESTATICO('Property','Value',...) creates a new
%     RESULTADOSDESP_EA_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before ResultadosDesp_EA_Estatico_OpeningFcn
%     gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to
%     ResultadosDesp_EA_Estatico_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
ResultadosDesp_EA_Estatico

% Last Modified by GUIDE v2.5 14-Nov-2011 18:12:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @ResultadosDesp_EA_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',   @ResultadosDesp_EA_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',   []);
```

```
@ResultadosDesp_EA_Estatico_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ResultadosDesp_EA_Estatico is made visible.
function ResultadosDesp_EA_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to ResultadosDesp_EA_Estatico (see
VARARGIN)

% Choose default command line output for ResultadosDesp_EA_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes ResultadosDesp_EA_Estatico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ResultadosDesp_EA_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global DesplazamientosEstaticoEA numero_nodos RestriccionesNodo
%Se crea este bucle para recoger en el vector matrizindicativaR unos
%valores que informaran al usuario que nodo y que desplazamiento es el
que
%se produce inmediatamente a la derecha de su valor
contadorEAed=0; %contador estructura articulada estatico
desplazamientos
for i=1:numero_nodos
    for I=1:2
        if (RestriccionesNodo(i,I)==1) && (I==1) && (contadorEAed==0)
            contadorEAed=contadorEAed+1;
            matrizindicativa=(10*i)+1;
        elseif (RestriccionesNodo(i,I)==1) && (I==2) &&
(contadorEAed==0)
            contadorEAed=contadorEAed+1;
            matrizindicativa=(10*i)+2;
```

```
        elseif (RestriccionesNodo(i,I)==1) && (I==1) &&
(contadorEAed~=0)
            contadorEAed=contadorEAed+1;
            matrizindicativa(contadorEAed,:)=(10*i)+1;
        elseif (RestriccionesNodo(i,I)==1) && (I==2) &&
(contadorEAed~=0)
            contadorEAed=contadorEAed+1;
            matrizindicativa(contadorEAed,:)=(10*i)+2;
        elseif (RestriccionesNodo(i,I)==0)
            contadorEAed=contadorEAed;
    end
end
end

DesplazamientosEstaticoEAtable=[matrizindicativa
DesplazamientosEstaticoEA];
%se muestra una tabla con los valres de los desplazamientos de los
nodos
set(handles.uitable3,'data',DesplazamientosEstaticoEAtable)

function desplazamientosEAe_output_Callback(hObject, eventdata,
handles)
% hObject    handle to desplazamientosEAe_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
desplazamientosEAe_output as text
%         str2double(get(hObject,'String')) returns contents of
desplazamientosEAe_output as a double

% --- Executes during object creation, after setting all properties.
function desplazamientosEAe_output_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to desplazamientosEAe_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function nododesplazazado_output_Callback(hObject, eventdata, handles)
% hObject    handle to nododesplazazado_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
nododesplazazado_output as text
%         str2double(get(hObject,'String')) returns contents of
nododesplazazado_output as a double
```

```
% --- Executes during object creation, after setting all properties.
function nododesplazado_output_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to nododesplazado_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana ResultadosDesp_EA_Estatico
close ResultadosDesp_EA_Estatico
```

### Ventana de resultados de fuerzas estáticas en estructuras reticuladas

```
function varargout = Calculo_Fuerzas_ER_Estatico(varargin)
% CALCULO_FUERZAS_ER_ESTATICO M-file for
% Calculo_Fuerzas_ER_Estatico.fig
%     CALCULO_FUERZAS_ER_ESTATICO, by itself, creates a new
%     CALCULO_FUERZAS_ER_ESTATICO or raises the existing
%     singleton*.
%
%     H = CALCULO_FUERZAS_ER_ESTATICO returns the handle to a new
%     CALCULO_FUERZAS_ER_ESTATICO or the handle to
%     the existing singleton*.
%
%
%     CALCULO_FUERZAS_ER_ESTATICO('CALLBACK',hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in CALCULO_FUERZAS_ER_ESTATICO.M with
%     the given input arguments.
%
%     CALCULO_FUERZAS_ER_ESTATICO('Property','Value',...) creates a
%     new CALCULO_FUERZAS_ER_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before
%     Calculo_Fuerzas_ER_Estatico_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to
%     Calculo_Fuerzas_ER_Estatico_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help
Calculo_Fuerzas_ER_Estatico

% Last Modified by GUIDE v2.5 20-Nov-2011 11:29:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Calculo_Fuerzas_ER_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',   @Calculo_Fuerzas_ER_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Calculo_Fuerzas_ER_Estatico is made
visible.
function Calculo_Fuerzas_ER_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Calculo_Fuerzas_ER_Estatico
(see VARARGIN)

% Choose default command line output for Calculo_Fuerzas_ER_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Calculo_Fuerzas_ER_Estatico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Calculo_Fuerzas_ER_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Se cierra la ventana Calculo_Fuerzas_ER_Estatico

close Calculo_Fuerzas_ER_Estatico

% --- Executes on button press in resultadosnumericos.
function resultadosnumericos_Callback(hObject, eventdata, handles)
% hObject    handle to resultadosnumericos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global E A I BARRAS Nodo RestriccionesNodo FuerzaNodos Barramuelle
ResultadosNumericosERe ans

%Se calculan las reacciones a partir de la funcion
%ReaccionesenestaticoestructuraREticulada.m y a partir de los datos
que
%estan entre parentesis
ReaccionesenestaticoestructuraREticulada(E,A,I,BARRAS,Nodo,Restriccion
esNodo,FuerzaNodos,Barramuelle);

%Los resultados obtenidos se guardan en la variable
ResultadosNumericosERe
ResultadosNumericosERe=ans;

%Se abre la ventana Resultados_Fuerzas_ER_Estatico
Resultados_Fuerzas_ER_Estatico

%Se espera a que se cierre la ventana Resultados_Fuerzas_ER_Estatico

uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject    handle to eliminar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Se eliminan todos los datos introducidos y calculados en el interfaz
y
%con el comando clc se limpia la ventana de comandos

clear all
clc
```

### Ventana de resultados de fuerzas estáticas en estructuras articuladas

```
function varargout = Calculo_Fuerzas_EA_Estatico(varargin)
% CALCULO_FUERZAS_EA_ESTATICO M-file for
% Calculo_Fuerzas_EA_Estatico.fig
%     CALCULO_FUERZAS_EA_ESTATICO, by itself, creates a new
%     CALCULO_FUERZAS_EA_ESTATICO or raises the existing
%     singleton*.
%
%     H = CALCULO_FUERZAS_EA_ESTATICO returns the handle to a new
%     CALCULO_FUERZAS_EA_ESTATICO or the handle to
%     the existing singleton*.
%
%
%     CALCULO_FUERZAS_EA_ESTATICO('CALLBACK',hObject,eventData,handles,...)
```

```
calls the local
%      function named CALLBACK in CALCULO_FUERZAS_EA_ESTATICO.M with
the given input arguments.
%
%      CALCULO_FUERZAS_EA_ESTATICO('Property','Value',...) creates a
new CALCULO_FUERZAS_EA_ESTATICO or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before
Calculo_Fuerzas_EA_Estatico_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Calculo_Fuerzas_EA_Estatico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Calculo_Fuerzas_EA_Estatico

% Last Modified by GUIDE v2.5 20-Nov-2011 11:28:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Calculo_Fuerzas_EA_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',  @Calculo_Fuerzas_EA_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Calculo_Fuerzas_EA_Estatico is made
visible.
function Calculo_Fuerzas_EA_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Calculo_Fuerzas_EA_Estatico
(see VARARGIN)

% Choose default command line output for Calculo_Fuerzas_EA_Estatico
handles.output = hObject;

% Update handles structure
```



```
guidata(hObject, handles);

% UIWAIT makes Calculo_Fuerzas_EA_Estatico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Calculo_Fuerzas_EA_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in calculreaccioA_output.
function calculreaccioA_output_Callback(hObject, eventdata, handles)
% hObject      handle to calculreaccioA_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana Calculo_Fuerzas_EA_Estatico
close Calculo_Fuerzas_EA_Estatico

% --- Executes on button press in resulnumA_output.
function resulnumA_output_Callback(hObject, eventdata, handles)
% hObject      handle to resulnumA_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global E A BARRAS Nodo RestriccionesNodo FuerzaNodos Barramuelle
ResultadosNumericosEAe ans
%Se calculan las reacciones a partir de la funcion
%Reaccionesenestaticoestructuraarticulada.m y a partir de los datos
que
%están entre parentesis
Reaccionesenestaticoestructuraarticulada(E,A,BARRAS,Nodo,Restricciones
Nodo,FuerzaNodos,Barramuelle);
%Los resultados obtenidos se guardan en la variable
ResultadosNumericosEAe

ResultadosNumericosEAe=ans;
%Se abre la ventana Resultados_Fuerzas_EA_Estatico

Resultados_Fuerzas_EA_Estatico

%Se espera a que se cierre la ventana Resultados_Fuerzas_EA_Estatico

uiwait
```

```
% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se eliminan todos los datos introducidos y calculados en el interfaz
Y
%con el comando clc se limpia la ventana de comandos

clear all
clc
```

### Ventana de resultados de fuerzas estáticas en estructuras articuladas

```
function varargout = Calculo_Fuerzas_EA_Estatico(varargin)
% CALCULO_FUERZAS_EA_ESTATICO M-file for
% Calculo_Fuerzas_EA_Estatico.fig
%     CALCULO_FUERZAS_EA_ESTATICO, by itself, creates a new
%     CALCULO_FUERZAS_EA_ESTATICO or raises the existing
%     singleton*.
%
%     H = CALCULO_FUERZAS_EA_ESTATICO returns the handle to a new
%     CALCULO_FUERZAS_EA_ESTATICO or the handle to
%     the existing singleton*.
%
%
%     CALCULO_FUERZAS_EA_ESTATICO('CALLBACK',hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in CALCULO_FUERZAS_EA_ESTATICO.M with
%     the given input arguments.
%
%     CALCULO_FUERZAS_EA_ESTATICO('Property','Value',...) creates a
%     new CALCULO_FUERZAS_EA_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before
%     Calculo_Fuerzas_EA_Estatico_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to
%     Calculo_Fuerzas_EA_Estatico_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
% Calculo_Fuerzas_EA_Estatico

% Last Modified by GUIDE v2.5 20-Nov-2011 11:28:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Calculo_Fuerzas_EA_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',   @Calculo_Fuerzas_EA_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
```

```
        'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Calculo_Fuerzas_EA_Estatico is made
% visible.
function Calculo_Fuerzas_EA_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Calculo_Fuerzas_EA_Estatico
% (see VARARGIN)

% Choose default command line output for Calculo_Fuerzas_EA_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Calculo_Fuerzas_EA_Estatico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Calculo_Fuerzas_EA_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in calculreaccioA_output.
function calculreaccioA_output_Callback(hObject, eventdata, handles)
% hObject    handle to calculreaccioA_output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Se cierra la ventana Calculo_Fuerzas_EA_Estatico
close Calculo_Fuerzas_EA_Estatico
```

```
% --- Executes on button press in resulnumA_output.
function resulnumA_output_Callback(hObject, eventdata, handles)
% hObject      handle to resulnumA_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global E A BARRAS Nodo RestriccionesNodo FuerzaNodos Barramuelle
ResultadosNumericosEAe ans
%Se calculan las reacciones a partir de la funcion
%Reaccionesenestaticoestructuraarticulada.m y a partir de los datos
que
%están entre parentesis
Reaccionesenestaticoestructuraarticulada(E,A,BARRAS,Nodo,Restricciones
Nodo,FuerzaNodos,Barramuelle);
%Los resultados obtenidos se guardan en la variable
ResultadosNumericosEAe

ResultadosNumericosEAe=ans;
%Se abre la ventana Resultados_Fuerzas_EA_Estatico

Resultados_Fuerzas_EA_Estatico

%Se espera a que se cierre la ventana Resultados_Fuerzas_EA_Estatico

uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se eliminan todos los datos introducidos y calculados en el interfaz
y
%con el comando clc se limpia la ventana de comandos

clear all
clc
```

### Resultados numéricos de fuerzas estáticas en estructuras reticuladas

```
function varargout = Resultados_Fuerzas_ER_Estatico(varargin)
% RESULTADOS_FUERZAS_ER_ESTATICO M-file for
Resultados_Fuerzas_ER_Estatico.fig
%     RESULTADOS_FUERZAS_ER_ESTATICO, by itself, creates a new
RESULTADOS_FUERZAS_ER_ESTATICO or raises the existing
%     singleton*.
%
%     H = RESULTADOS_FUERZAS_ER_ESTATICO returns the handle to a new
RESULTADOS_FUERZAS_ER_ESTATICO or the handle to
%     the existing singleton*.
%
%
RESULTADOS_FUERZAS_ER_ESTATICO('CALLBACK',hObject,eventData,handles,...
.) calls the local
%     function named CALLBACK in RESULTADOS_FUERZAS_ER_ESTATICO.M
with the given input arguments.
%
%     RESULTADOS_FUERZAS_ER_ESTATICO('Property','Value',...) creates
a new RESULTADOS_FUERZAS_ER_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
```

```
pairs are
%      applied to the GUI before
Resultados_Fuerzas_ER_Estatico_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Resultados_Fuerzas_ER_Estatico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Resultados_Fuerzas_ER_Estatico

% Last Modified by GUIDE v2.5 14-Nov-2011 20:20:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Resultados_Fuerzas_ER_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',    @Resultados_Fuerzas_ER_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Resultados_Fuerzas_ER_Estatico is made
visible.
function Resultados_Fuerzas_ER_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Resultados_Fuerzas_ER_Estatico
(see VARARGIN)

% Choose default command line output for
Resultados_Fuerzas_ER_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Resultados_Fuerzas_ER_Estatico wait for user response
(see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = Resultados_Fuerzas_ER_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global ResultadosNumericosERe numero_nodos
%se crea un vector que servira para indicar el nodo y el
desplazamiento de
%los valores que aparezca inmediatamente a la izquierda de ellos
matrizdedieices=ones(numero_nodos,3)*10;
contador=0;
for i=1:numero_nodos
    for I=1:3
        if (I==1)
            contador=contador+1;
            columnaforcesas(contador,:)=(matrizdedieices(i,I)*i)+1;
        elseif (I==2)
            contador=contador+1;
            columnaforcesas(contador,:)=(matrizdedieices(i,I)*i)+2;
        elseif (I==3)
            contador=contador+1;
            columnaforcesas(contador,:)=(matrizdedieices(i,I)*i)+3;
        end
    end
end

ResultadosNumericosERestatico=[columnaforcesas ResultadosNumericosERe];
%Se muestra en pantalla una tabla con los resultados numericos de las
reacciones y fuerzas de los nodos de la estructura
set(handles.uitable1,'data',ResultadosNumericosERestatico)

function resultados_output_Callback(hObject, eventdata, handles)
% hObject      handle to resultados_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of resultados_output
as text
%          str2double(get(hObject,'String')) returns contents of
resultados_output as a double

% --- Executes during object creation, after setting all properties.
function resultados_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to resultados_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

end

```
function columnanodos_output_Callback(hObject, eventdata, handles)
% hObject      handle to columnanodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of columnanodos_output
as text
%          str2double(get(hObject,'String')) returns contents of
columnanodos_output as a double

% --- Executes during object creation, after setting all properties.
function columnanodos_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to columnanodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana Resultados_Fuerzas_ER_Estatico
close Resultados_Fuerzas_ER_Estatico
```

### Resultados numéricos de fuerzas estáticas en estructuras articuladas

```
function varargout = Resultados_Fuerzas_EA_Estatico(varargin)
% RESULTADOS_FUERZAS_EA_ESTATICO M-file for
Resultados_Fuerzas_EA_Estatico.fig
%     RESULTADOS_FUERZAS_EA_ESTATICO, by itself, creates a new
RESULTADOS_FUERZAS_EA_ESTATICO or raises the existing
%     singleton*.
%
%     H = RESULTADOS_FUERZAS_EA_ESTATICO returns the handle to a new
RESULTADOS_FUERZAS_EA_ESTATICO or the handle to
%     the existing singleton*.
%
%
RESULTADOS_FUERZAS_EA_ESTATICO('CALLBACK',hObject,eventData,handles,...
.) calls the local
%     function named CALLBACK in RESULTADOS_FUERZAS_EA_ESTATICO.M
with the given input arguments.
%
%     RESULTADOS_FUERZAS_EA_ESTATICO('Property','Value',...) creates
a new RESULTADOS_FUERZAS_EA_ESTATICO or raises the
%     existing singleton*. Starting from the left, property value
pairs are
```

```
%      applied to the GUI before
Resultados_Fuerzas_EA_Estatico_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Resultados_Fuerzas_EA_Estatico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Resultados_Fuerzas_EA_Estatico

% Last Modified by GUIDE v2.5 14-Nov-2011 20:49:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Resultados_Fuerzas_EA_Estatico_OpeningFcn, ...
                  'gui_OutputFcn',   @Resultados_Fuerzas_EA_Estatico_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Resultados_Fuerzas_EA_Estatico is made
visible.
function Resultados_Fuerzas_EA_Estatico_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Resultados_Fuerzas_EA_Estatico
(see VARARGIN)

% Choose default command line output for
Resultados_Fuerzas_EA_Estatico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Resultados_Fuerzas_EA_Estatico wait for user response
(see UIRESUME)
% uiwait(handles.figure1);
```



```
% --- Outputs from this function are returned to the command line.
function varargout = Resultados_Fuerzas_EA_Estatico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global ResultadosNumericosEAe numero_nodos
%se crea un vector que servira para indicar el nodo y el
desplazamiento de
%los valores que aparezca inmediatamente a la izquierda de ellos
matrizdedieices=ones(numero_nodos,2)*10;
contador=0;
for i=1:numero_nodos
    for I=1:2
        if (I==1)
            contador=contador+1;
            columnaforcesas(contador,:)=(matrizdedieices(i,I)*i)+1;
        elseif (I==2)
            contador=contador+1;
            columnaforcesas(contador,:)=(matrizdedieices(i,I)*i)+2;
        elseif (I==3)
            % contador=contador+1;
            % columnaforcesas(contador,:)=matrizdedieices(i,I)+3;
        end
    end
end

ResultadosNumericosEAestatico=[columnaforcesas ResultadosNumericosEAe];
%Se muestra en pantalla una tabla con los resultados numericos de las
reacciones y fuerzas de los nodos de la estructura

set(handles.uitable1,'data',ResultadosNumericosEAestatico)

function resultados_output_Callback(hObject, eventdata, handles)
% hObject      handle to resultados_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of resultados_output
as text
%          str2double(get(hObject,'String')) returns contents of
resultados_output as a double

% --- Executes during object creation, after setting all properties.
function resultados_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to resultados_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function columnanodos_output_Callback(hObject, eventdata, handles)
% hObject      handle to columnanodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of columnanodos_output
as text
%      str2double(get(hObject,'String')) returns contents of
columnanodos_output as a double

% --- Executes during object creation, after setting all properties.
function columnanodos_output_CreateFcn(hObject, eventdata, handles)
% hObject      handle to columnanodos_output (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana Resultados_Fuerzas_EA_Estatico
close Resultados_Fuerzas_EA_Estatico
```

### Ventana de resultados de desplazamientos en dinámico en estructura reticuladas

```
function varargout = Desplazamiento_Reticuladas(varargin)
% DESPLAZAMIENTO_RETICULADAS M-file for Desplazamiento_Reticuladas.fig
%   DESPLAZAMIENTO_RETICULADAS, by itself, creates a new
DESPLAZAMIENTO_RETICULADAS or raises the existing
%   singleton*.
%
%   H = DESPLAZAMIENTO_RETICULADAS returns the handle to a new
DESPLAZAMIENTO_RETICULADAS or the handle to
%   the existing singleton*.
%
%
DESPLAZAMIENTO_RETICULADAS('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in DESPLAZAMIENTO_RETICULADAS.M with
the given input arguments.
%
%   DESPLAZAMIENTO_RETICULADAS('Property','Value',...) creates a
new DESPLAZAMIENTO_RETICULADAS or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before Desplazamiento_Reticuladas_OpeningFcn
```

```
gets called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to
Desplazamiento_Reticuladas_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Desplazamiento_Reticuladas

% Last Modified by GUIDE v2.5 20-Nov-2011 11:34:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Desplazamiento_Reticuladas_OpeningFcn, ...
                  'gui_OutputFcn',   @Desplazamiento_Reticuladas_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Desplazamiento_Reticuladas is made visible.
function Desplazamiento_Reticuladas_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Desplazamiento_Reticuladas (see
VARARGIN)

% Choose default command line output for Desplazamiento_Reticuladas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Desplazamiento_Reticuladas wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Desplazamiento_Reticuladas_OutputFcn(hObject,
eventdata, handles)
```

```
% varargin    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject     handle to calcular (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Calculo de los desplazamientos a partir de la funcion
%xprimaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ, mediante
la
%instruccion ode45 y a partir de los datos que aparecen entre
parentesis

global E A I densidad B omegaaa BARRAS Nodo Barramuelle
Barraamortiguador FuerzaNodos RestriccionesNodo t x tini tfin
condicionesiniciales desfase
[t,x]= ode45 (@xprimareticuladaconcoordenadasymuellePRUEBA,
[tini,tfin],condicionesiniciales,[],E,A,I,densidad,B,omegaaa,BARRAS,No
do,Barramuelle,Barraamortiguador,FuerzaNodos,RestriccionesNodo,tfin,de
sfase);

% --- Executes on button press in graficasdesptiem.
function graficasdesptiem_Callback(hObject, eventdata, handles)
% hObject     handle to graficasdesptiem (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%se abre la ventana ResolucionGraficasr
ResolucionGraficasr
%se mantiene en espera hasta que se cierre la ventana
ResolucionGraficasr
uiwait

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject     handle to cerrar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%se cierra la ventana Desplazamiento_Reticuladas
close Desplazamiento_Reticuladas

% --- Executes on button press in
resultadosnume_Desplazamiento_Reticuladas.
function resultadosnume_Desplazamiento_Reticuladas_Callback(hObject,
eventdata, handles)
% hObject     handle to resultadosnume_Desplazamiento_Reticuladas (see
GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Se abre la ventana Resultados_desplazamientos_dinamico_Reticuladas
```

```
Resultados_desplazamientos_dinamico_Reticuladas
%Se mantiene en espera hasta que se cierre la ventana
%Resultados_desplazamientos_dinamico_Reticuladas
uiwait
```

```
% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject handle to eliminar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%se eliminan los datos introducidos y calculados y se limpia la
ventana de
%comandos
clear all
clc
```

### Ventana de resultados de desplazamientos en dinámico en estructura articuladas

```
function varargout = Desplazamiento_Articualdas(varargin)
% DESPLAZAMIENTO_ARTICUALDAS M-file for Desplazamiento_Articualdas.fig
% DESPLAZAMIENTO_ARTICUALDAS, by itself, creates a new
DESPLAZAMIENTO_ARTICUALDAS or raises the existing
% singleton*.
%
% H = DESPLAZAMIENTO_ARTICUALDAS returns the handle to a new
DESPLAZAMIENTO_ARTICUALDAS or the handle to
% the existing singleton*.
%
%
DESPLAZAMIENTO_ARTICUALDAS('CALLBACK',hObject,eventData,handles,...)
calls the local
% function named CALLBACK in DESPLAZAMIENTO_ARTICUALDAS.M with
the given input arguments.
%
% DESPLAZAMIENTO_ARTICUALDAS('Property','Value',...) creates a
new DESPLAZAMIENTO_ARTICUALDAS or raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before Desplazamiento_Articualdas_OpeningFcn
gets called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to
Desplazamiento_Articualdas_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Desplazamiento_Articualdas

% Last Modified by GUIDE v2.5 20-Nov-2011 11:31:38
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Desplazamiento_Articualdas_OpeningFcn, ...
                  'gui_OutputFcn',    @Desplazamiento_Articualdas_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Desplazamiento_Articualdas is made visible.
function Desplazamiento_Articualdas_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Desplazamiento_Articualdas (see
VARARGIN)

% Choose default command line output for Desplazamiento_Articualdas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Desplazamiento_Articualdas wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Desplazamiento_Articualdas_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

%Calculo de los desplazamientos a partir de la funcion
%primaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ, mediante
```

```
la
%instruccion ode45 y a partir de los datos que aparecen entre
parentesis

global E A densidad omegaaa B BARRAS Nodo RestriccionesNodo
FuerzaNodos Barramuelle Barraamortiguador condicionesiniciales t x
tini tfin desfase
[t,x]= ode45
(@xprimaarticuladaconcoordenadasmuelleyamortiguadorINTERFAZ,
[tini,tfin],condicionesiniciales,[],E,A,densidad,omegaaa,B,BARRAS,Nodo
,RestriccionesNodo,FuerzaNodos,Barramuelle,Barraamortiguador,tfin,desf
ase);

% --- Executes on button press in
graficasdesplazamientotiempo_menuppal.
function graficasdesplazamientotiempo_menuppal_Callback(hObject,
eventdata, handles)
% hObject    handle to graficasdesplazamientotiempo_menuppal (see
GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se abre la ventana ResolucionGraficas
ResolucionGraficas
%se espera hasta que la ventana ResolucionGraficas sea cerrada
uiwait

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se cierra la ventana Desplazamiento_Articualdas
close Desplazamiento_Articualdas

% --- Executes on button press in
resultadosnumericos_Desplazamiento_Articuladas.
function
resultadosnumericos_Desplazamiento_Articuladas_Callback(hObject,
eventdata, handles)
% hObject    handle to resultadosnumericos_Desplazamiento_Articuladas
(see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se abre la ventana Resultados_Numericos_Articuladas_Dinamico
Resultados_Numericos_Articuladas_Dinamico

%se espera hasta que la ventana
Resultados_Numericos_Articuladas_Dinamico
%sea cerrada

uiwait
```

```
% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%se eliminan los datos introducidos y calculados y se limpia la
ventana de
%comandos

clear all
clc
```

## Resultados numéricos de los desplazamientos dinámicos en estructuras reticuladas

```
function varargout =
Resultados_desplazamientos_dinamico_Reticuladas(varargin)
% RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS M-file for
Resultados_desplazamientos_dinamico_Reticuladas.fig
%     RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS, by itself,
creates a new RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS or
raises the existing
%     singleton*.
%
%     H = RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS returns the
handle to a new RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS or the
handle to
%     the existing singleton*.
%
%
RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS('CALLBACK',hObject,eve
ntData,handles,...) calls the local
%     function named CALLBACK in
RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS.M with the given input
arguments.
%
%
RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS('Property','Value',...
) creates a new RESULTADOS_DESPLAZAMIENTOS_DINAMICO_RETICULADAS or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Resultados_desplazamientos_dinamico_Reticuladas_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Resultados_desplazamientos_dinamico_Reticuladas_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Resultados_desplazamientos_dinamico_Reticuladas

% Last Modified by GUIDE v2.5 19-Nov-2011 21:16:58
```



```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Resultados_desplazamientos_dinamico_Reticuladas_OpeningFcn, ...
                  'gui_OutputFcn',    @Resultados_desplazamientos_dinamico_Reticuladas_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before
Resultados_desplazamientos_dinamico_Reticuladas is made visible.
function
Resultados_desplazamientos_dinamico_Reticuladas_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to
Resultados_desplazamientos_dinamico_Reticuladas (see VARARGIN)

% Choose default command line output for
Resultados_desplazamientos_dinamico_Reticuladas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Resultados_desplazamientos_dinamico_Reticuladas wait
for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout =
Resultados_desplazamientos_dinamico_Reticuladas_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global numero_nodos t x RestriccionesNodo

%Estos bucles permiten obtener el vector cnames que se le da al usuario
para
```

```
%comprender a que nodo corresponde cada desplazamientos

contador2=0;
for i=1:numero_nodos
    for I=1:3
        if (RestriccionesNodo(i,I)==1)
            contador2=contador2+1;
        end
    end
end

contador=0;
for i=1:numero_nodos
    for I=1:3
        if RestriccionesNodo(i,I)==0
            contador=contador;
        elseif (RestriccionesNodo(i,I)==1) && (I==1)
            contador=contador+1;
            cnames(1,contador)=1+(10*i);
            cnames(1,contador+contador2)=1+(100*i);
        elseif (RestriccionesNodo(i,I)==1) && (I==2)
            contador=contador+1;
            cnames(1,contador)=2+(10*i);
            cnames(1,contador+contador2)=2+(100*i);
        elseif (RestriccionesNodo(i,I)==1) && (I==3)
            contador=contador+1;
            cnames(1,contador)=3+(10*i);
            cnames(1,contador+contador2)=3+(100*i);
        end
    end
end

cnames={0 cnames};
resultados=[t x];

%Se muestran en una tabla los resultados

set(handles.uitable1,'data',resultados)
%se le asigna como titulo de columna los valores del vector cnames
set(handles.uitable1,'ColumnName',cnames)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana Resultados_desplazamientos_dinamico_Reticuladas
close Resultados_desplazamientos_dinamico_Reticuladas
```

### Resultados numéricos de los desplazamientos en dinámicos en estructuras articuladas

```
function varargout =
Resultados_Numericos_Articuladas_Dinamico(varargin)
% RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO M-file for
Resultados_Numericos_Articuladas_Dinamico.fig
%     RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO, by itself, creates a
new RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO or raises the existing
%     singleton*.
```

```
%
%      H = RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO returns the
handle to a new RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO or the
handle to
%      the existing singleton*.
%
%
RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO('CALLBACK',hObject,eventData
,handles,...) calls the local
%      function named CALLBACK in
RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO.M with the given input
arguments.
%
%
RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO('Property','Value',...)
creates a new RESULTADOS_NUMERICOS_ARTICULADAS_DINAMICO or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before
Resultados_Numericos_Articuladas_Dinamico_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Resultados_Numericos_Articuladas_Dinamico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Resultados_Numericos_Articuladas_Dinamico

% Last Modified by GUIDE v2.5 19-Nov-2011 21:24:02

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Resultados_Numericos_Articuladas_Dinamico_OpeningFcn, ...
                  'gui_OutputFcn',    @Resultados_Numericos_Articuladas_Dinamico_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Resultados_Numericos_Articuladas_Dinamico
is made visible.
function Resultados_Numericos_Articuladas_Dinamico_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to
Resultados_Numericos_Articuladas_Dinamico (see VARARGIN)

% Choose default command line output for
Resultados_Numericos_Articuladas_Dinamico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Resultados_Numericos_Articuladas_Dinamico wait for user
response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout =
Resultados_Numericos_Articuladas_Dinamico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global t x numero_nodos RestriccionesNodo

%Estos bucles permiten obtener el vector cnames que se le da al usuario
para
%comprender a que nodo corresponde cada desplazamiento

contador2=0;
for i=1:numero_nodos
    for I=1:2
        if RestriccionesNodo(i,I)~=0
            contador2=contador2+1;
        end
    end
end

contador=0;
for i=1:numero_nodos
    for I=1:2
        if RestriccionesNodo(i,I)==0
            contador=contador;
        elseif (RestriccionesNodo(i,I)==1) && (I==1)
            contador=contador+1;
            cnames(1,contador)=1+(10*i);
            cnames(1,contador+contador2)=1+(100*i);
        elseif (RestriccionesNodo(i,I)==1) && (I==2)
            contador=contador+1;
            cnames(1,contador)=2+(10*i);
            cnames(1,contador+contador2)=2+(100*i);
        end
    end
end
```

```
cnames={0 cnames};
resultadosdinamicoarticuladas=[t x];

%Se muestran en una tabla los resultados
set(handles.resultados_output,'data',resultadosdinamicoarticuladas)
%se le asigna como titulo de columna los valores del vector cnames

set(handles.resultados_output,'ColumnName',cnames)

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana Resultados_Numericos_Articuladas_Dinamico
close Resultados_Numericos_Articuladas_Dinamico
```

### Resultados en gráficas de los desplazamientos en dinámico de estructuras reticuladas

```
function varargout = ResolucionGraficasr(varargin)
% RESOLUCIONGRAFICASR M-file for ResolucionGraficasr.fig
%   RESOLUCIONGRAFICASR, by itself, creates a new
%   RESOLUCIONGRAFICASR or raises the existing
%   singleton*.
%
%   H = RESOLUCIONGRAFICASR returns the handle to a new
%   RESOLUCIONGRAFICASR or the handle to
%   the existing singleton*.
%
%   RESOLUCIONGRAFICASR('CALLBACK',hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in RESOLUCIONGRAFICASR.M with the given
%   input arguments.
%
%   RESOLUCIONGRAFICASR('Property','Value',...) creates a new
%   RESOLUCIONGRAFICASR or raises the
%   existing singleton*. Starting from the left, property value
%   pairs are
%   applied to the GUI before ResolucionGraficasr_OpeningFcn gets
%   called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to ResolucionGraficasr_OpeningFcn
%   via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%   only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
ResolucionGraficasr

% Last Modified by GUIDE v2.5 03-Nov-2011 15:46:55

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @ResolucionGraficasr_OpeningFcn,
...
'gui_OutputFcn', @ResolucionGraficasr_OutputFcn,
...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ResolucionGraficasr is made visible.
function ResolucionGraficasr_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to ResolucionGraficasr (see
VARARGIN)

% Choose default command line output for ResolucionGraficasr
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes ResolucionGraficasr wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ResolucionGraficasr_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

%Se cierra la ventana ResolucionGraficasr
close ResolucionGraficasr
```

```
function nodogra_input_Callback(hObject, eventdata, handles)
% hObject      handle to nodogra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodogra_input as
text
%           str2double(get(hObject,'String')) returns contents of
nodogra_input as a double

% --- Executes during object creation, after setting all properties.
function nodogra_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodogra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function colgra_input_Callback(hObject, eventdata, handles)
% hObject      handle to colgra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of colgra_input as
text
%           str2double(get(hObject,'String')) returns contents of
colgra_input as a double

% --- Executes during object creation, after setting all properties.
function colgra_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to colgra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ejegra_input_Callback(hObject, eventdata, handles)
% hObject      handle to ejegra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ejegra_input as
text
%           str2double(get(hObject,'String')) returns contents of
```

ejegra\_input as a double

```
% --- Executes during object creation, after setting all properties.
function ejegra_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ejegra_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global t x RestriccionesNodo numero_nodos

contador=0;
for i=1:numero_nodos
    for I=1:3
        if RestriccionesNodo(i,I)==0
            matdespl(i,I)=RestriccionesNodo(i,I);
        elseif RestriccionesNodo(i,I)==1
            contador=contador+1;
            matdespl(i,I)=RestriccionesNodo(i,I)*contador;
        end
    end
end
%se asigna el valor elegido por el usuario para el nodo en la variable
%nodoo
nodoo=str2double(get(handles.nodogra_input,'String'));
%se asigna el valor elegido por el usuario para el eje en la variable
%ejee
ejee=str2double(get(handles.ejegra_input,'String'));
%se asigna el valor elegido por el usuario para el color de la grafica
en la variable
%color
color=str2double(get(handles.colgra_input,'String'));
if color==1
    plot(t,x(:,matdespl(nodoo,ejee)),'g');
elseif color==2
    plot(t,x(:,matdespl(nodoo,ejee)),'r');
elseif color==3
    plot(t,x(:,matdespl(nodoo,ejee)),'c');
elseif color==4
    plot(t,x(:,matdespl(nodoo,ejee)),'m');
elseif color==5
    plot(t,x(:,matdespl(nodoo,ejee)),'y');
elseif color==6
    plot(t,x(:,matdespl(nodoo,ejee)),'k');
elseif color==7
    plot(t,x(:,matdespl(nodoo,ejee)),'--');
else
    plot(t,x(:,matdespl(nodoo,ejee)));
end
```



```
%Se titula la grafica, asi como el eje x y el eje y
title('Desplazamiento-Tiempo');
xlabel('Tiempo');
ylabel('Desplazamiento');
guidata(hObject, handles);

% --- Executes on button press in limpiar.
function limpiar_Callback(hObject, eventdata, handles)
% hObject    handle to limpiar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%se limpia el axes
cla(handles.axes1,'reset')
guidata(hObject, handles);

% --- Executes on button press in holdoff.
function holdoff_Callback(hObject, eventdata, handles)
% hObject    handle to holdoff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%las graficas dejan de superponerse
hold off

% --- Executes on button press in holdon.
function holdon_Callback(hObject, eventdata, handles)
% hObject    handle to holdon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Esta orden hace que la grafica dibujada en pantalla no se borre al
pedir
%que se dibuje una nueva, permitiendo comparar todas las graficas que
se
%desee
hold on

% --- Executes on button press in limpiarode.
function limpiarode_Callback(hObject, eventdata, handles)
% hObject    handle to limpiarode (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%se eliminan todos los datos introducidos y calculados
clear all
```

### Resultados en gráficas de los desplazamientos en dinámico de estructuras articuladas

```
function varargout = ResolucionGraficas(varargin)
% RESOLUCIONGRAFICAS M-file for ResolucionGraficas.fig
%     RESOLUCIONGRAFICAS, by itself, creates a new RESOLUCIONGRAFICAS
or raises the existing
%     singleton*.
%
%     H = RESOLUCIONGRAFICAS returns the handle to a new
RESOLUCIONGRAFICAS or the handle to
%     the existing singleton*.
%
%     RESOLUCIONGRAFICAS('CALLBACK',hObject,eventData,handles,...)
calls the local
```

```
%      function named CALLBACK in RESOLUCIONGRAFICAS.M with the given
input arguments.
%
%      RESOLUCIONGRAFICAS('Property','Value',...) creates a new
RESOLUCIONGRAFICAS or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before ResolucionGraficas_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to ResolucionGraficas_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
ResolucionGraficas

% Last Modified by GUIDE v2.5 02-Nov-2011 11:02:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ResolucionGraficas_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @ResolucionGraficas_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ResolucionGraficas is made visible.
function ResolucionGraficas_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to ResolucionGraficas (see
VARARGIN)

% Choose default command line output for ResolucionGraficas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes ResolucionGraficas wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ResolucionGraficas_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function nodogra_input_Callback(hObject, eventdata, handles)
% hObject      handle to nodogra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodogra_input as
text
%           str2double(get(hObject,'String')) returns contents of
nodogra_input as a double

% --- Executes during object creation, after setting all properties.
function nodogra_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodogra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function colgra_input_Callback(hObject, eventdata, handles)
% hObject      handle to colgra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of colgra_input as
text
%           str2double(get(hObject,'String')) returns contents of
colgra_input as a double

% --- Executes during object creation, after setting all properties.
function colgra_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to colgra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ejegra_input_Callback(hObject, eventdata, handles)
% hObject      handle to ejegra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ejegra_input as
text
%      str2double(get(hObject,'String')) returns contents of
ejegra_input as a double

% --- Executes during object creation, after setting all properties.
function ejegra_input_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ejegra_input (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global t x RestriccionesNodo numero_nodos

contador=0;
for i=1:numero_nodos
    for I=1:2
        if RestriccionesNodo(i,I)==0
            matdespl(i,I)=RestriccionesNodo(i,I);
        elseif RestriccionesNodo(i,I)==1
            contador=contador+1;
            matdespl(i,I)=RestriccionesNodo(i,I)*contador;
        end
    end
end
%memoriza en la variable nodoo el valor de nodo asignado por el
usuario
nodoo=str2double(get(handles.nodogra_input,'String'));
%memoriza en la variable ejee el valor asignado por el usuario
ejee=str2double(get(handles.ejegra_input,'String'));
%memoriza en la variable color el valor asignado por el usuario
color=str2double(get(handles.colgra_input,'String'));

%mediante este bucle se consigue que la grafica salga de un
determinado
```

```
%color
if color==1
plot(t,x(:,matdespl(nodoo,ejee)), 'g');
elseif color==2
plot(t,x(:,matdespl(nodoo,ejee)), 'r');
elseif color==3
plot(t,x(:,matdespl(nodoo,ejee)), 'c');
elseif color==4
plot(t,x(:,matdespl(nodoo,ejee)), 'm');
elseif color==5
plot(t,x(:,matdespl(nodoo,ejee)), 'y');
elseif color==6
plot(t,x(:,matdespl(nodoo,ejee)), 'k');
elseif color==7
plot(t,x(:,matdespl(nodoo,ejee)), '--');
else
plot(t,x(:,matdespl(nodoo,ejee)));
end

%Aqui se titula la gr·fica y se le da el nombre a los jees x e y.
title('Desplazamiento-Tiempo');
xlabel('Tiempo');
ylabel('Desplazamiento');
guidata(hObject, handles);

% --- Executes on button press in limpiar.
function limpiar_Callback(hObject, eventdata, handles)
% hObject      handle to limpiar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%se limpia el axes
cla(handles.axes1, 'reset')
guidata(hObject, handles);

% --- Executes on button press in holdoff.
function holdoff_Callback(hObject, eventdata, handles)
% hObject      handle to holdoff (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%mediante esta orden las graficas se dejan de superponer
hold off

% --- Executes on button press in holdon.
function holdon_Callback(hObject, eventdata, handles)
% hObject      handle to holdon (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Esta orden hace que la gr·fica dibujada en pantalla no se borre al
pedir
%que se dibuje una nueva, permitiendo comparar todas las graficas que
se
%desee
hold on

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
% hObject      handle to close (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%se cierra la ventana ResolucionGraficas
```

close ResolucionGraficas

```
% --- Executes on button press in limpiarode.  
function limpiarode_Callback(hObject, eventdata, handles)  
% hObject      handle to limpiarode (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
%se eliminan todos los datos introducidos y calculados  
clear all
```

## Ventana de resultados de fuerzas dinámicas en estructuras reticuladas

```
function varargout = FuerzasReticulada(varargin)  
% FUERZASRETICULADA M-file for FuerzasReticulada.fig  
%     FUERZASRETICULADA, by itself, creates a new FUERZASRETICULADA  
or raises the existing  
%     singleton*.  
%  
%     H = FUERZASRETICULADA returns the handle to a new  
FUERZASRETICULADA or the handle to  
%     the existing singleton*.  
%  
%     FUERZASRETICULADA('CALLBACK',hObject,eventData,handles,...)  
calls the local  
%     function named CALLBACK in FUERZASRETICULADA.M with the given  
input arguments.  
%  
%     FUERZASRETICULADA('Property','Value',...) creates a new  
FUERZASRETICULADA or raises the  
%     existing singleton*. Starting from the left, property value  
pairs are  
%     applied to the GUI before FuerzasReticulada_OpeningFcn gets  
called. An  
%     unrecognized property name or invalid value makes property  
application  
%     stop. All inputs are passed to FuerzasReticulada_OpeningFcn  
via varargin.  
%  
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows  
only one  
%     instance to run (singleton)".  
%  
% See also: GUIDE, GUIDATA, GUIHANDLES  
  
% Edit the above text to modify the response to help FuerzasReticulada  
  
% Last Modified by GUIDE v2.5 20-Nov-2011 11:37:10  
  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name',       mfilename, ...  
                  'gui_Singleton',  gui_Singleton, ...  
                  'gui_OpeningFcn', @FuerzasReticulada_OpeningFcn, ...  
                  'gui_OutputFcn',  @FuerzasReticulada_OutputFcn, ...  
                  'gui_LayoutFcn',  [] , ...  
                  'gui_Callback',   []);  
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before FuerzasReticulada is made visible.
function FuerzasReticulada_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to FuerzasReticulada (see
VARARGIN)

% Choose default command line output for FuerzasReticulada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FuerzasReticulada wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FuerzasReticulada_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

global E A densidad omegaaa B I BARRAS Nodo RestriccionesNodo
FuerzaNodos Barramuelle Barraamortiguador t x tini tfin ans
ReaccionesDinamicoReticulada
%Se obtienen las reacciones de una estructura articulada sometida a
fuerzas
%dinamicas a partir del fichero
reaccionesarticulconcoordenadasINTERFAZ.m y
%Este a partir de los datos introducidos en el parentesis
reaccionesreticulconcoordenadasINTERFAZ(E,A,densidad,x,BARRAS,Nodo,Res
triccionesNodo,Barraamortiguador,Barramuelle,I);
%Se guardan los resultados de las reacciones en la variable
%ReaccionesDinamicoArticuladas
ReaccionesDinamicoReticulada=ans;

% --- Executes on button press in graffuertiem.
```

```
function graffuertiem_Callback(hObject, eventdata, handles)
% hObject      handle to graffuertiem (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana GraficasFuerzTiempRet
GraficasFuerzTiempRet
%se mantiene en espera hasta que se cierra la ventana
GraficasFuerzTiempRet
uiwait

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana FuerzasReticulada
close FuerzasReticulada

% --- Executes on button press in resultadosnume_FuerzasReticuladas.
function resultadosnume_FuerzasReticuladas_Callback(hObject,
eventdata, handles)
% hObject      handle to resultadosnume_FuerzasReticuladas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se abre la ventana ReaccionesER_Dinamico
ReaccionesER_Dinamico
%se mantiene en espera hasta que se cierra la ventana
ReaccionesER_Dinamico
uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject      handle to eliminar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%se eliminan todos los datos introducidos y calculados y se limpia la
%ventana de comandos

clear all
clc
```

### Ventana de resultados de fuerzas dinámicas en estructuras articuladas

```
function varargout = FuerzasArticuladas(varargin)
% FUERZASARTICULADAS M-file for FuerzasArticuladas.fig
%   FUERZASARTICULADAS, by itself, creates a new FUERZASARTICULADAS
or raises the existing
%   singleton*.
%
%   H = FUERZASARTICULADAS returns the handle to a new
FUERZASARTICULADAS or the handle to
%   the existing singleton*.
%
%   FUERZASARTICULADAS('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in FUERZASARTICULADAS.M with the given
```



```
input arguments.
%
%     FUERZASARTICULADAS('Property','Value',...) creates a new
FUERZASARTICULADAS or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before FuerzasArticuladas_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to FuerzasArticuladas_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
FuerzasArticuladas

% Last Modified by GUIDE v2.5 20-Nov-2011 11:35:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @FuerzasArticuladas_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @FuerzasArticuladas_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before FuerzasArticuladas is made visible.
function FuerzasArticuladas_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to FuerzasArticuladas (see
VARARGIN)

% Choose default command line output for FuerzasArticuladas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FuerzasArticuladas wait for user response (see
```

```
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FuerzasArticuladas_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject      handle to calcular (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global E A densidad omegaaa B BARRAS Nodo RestriccionesNodo
FuerzaNodos Barramuelle Barraamortiguador t x tini tfin ans
ReaccionesDinamicoArticuladas

%Se obtienen las reacciones de una estructura articulada sometida a
fuerzas
%dinamicas a partir del fichero
reaccionesarticulconcoordenadasINTERFAZ.m y
%Èste a partir de los datos introducidos en el parentesis
reaccionesarticulconcoordenadasINTERFAZ(E,A,densidad,x,BARRAS,Nodo,Res
triccionesNodo,Barraamortiguador,Barramuelle);

%Se guardan los resultados de las reacciones en la variable
%ReaccionesDinamicoArticuladas
ReaccionesDinamicoArticuladas=ans;

% --- Executes on button press in graffuertiem.
function graffuertiem_Callback(hObject, eventdata, handles)
% hObject      handle to graffuertiem (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%global numero_nodos t ans

%Se abre la ventana GraficasFuerzTiemp
GraficasFuerzTiemp
%se mantiene en espera hasta que se cierra la ventana
GraficasFuerzTiemp
uiwait

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Se cierra la ventana FuerzasArticuladas
```

```
close FuerzasArticuladas

% --- Executes on button press in
resultadosnumericos_FuerzasArticuladas.
function resultadosnumericos_FuerzasArticuladas_Callback(hObject,
eventdata, handles)
% hObject    handle to resultadosnumericos_FuerzasArticuladas (see
GCB0)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Se abre la ventana Resultados_Reacciones_Dinamico_Articuladas

Resultados_Reacciones_Dinamico_Articuladas
%Se mantiene en espera hasta que se cierra la ventana
Resultados_Reacciones_Dinamico_Articuladas

uiwait

% --- Executes on button press in eliminar.
function eliminar_Callback(hObject, eventdata, handles)
% hObject    handle to eliminar (see GCB0)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Se eliminan todos los datos introducidos y calculados, y ademas, se
limpia
%la ventana de comandos

clear all
clc
```

## Resultados en gráficas de fuerzas dinámicas en estructuras reticuladas

```
function varargout = GraficasFuerzTiempRet(varargin)
% GRAFICASFUERZTIEMPRET M-file for GraficasFuerzTiempRet.fig
%   GRAFICASFUERZTIEMPRET, by itself, creates a new
GRAFICASFUERZTIEMPRET or raises the existing
%   singleton*.
%
%   H = GRAFICASFUERZTIEMPRET returns the handle to a new
GRAFICASFUERZTIEMPRET or the handle to
%   the existing singleton*.
%
%   GRAFICASFUERZTIEMPRET('CALLBACK',hObject,eventData,handles,...)
calls the local
%   function named CALLBACK in GRAFICASFUERZTIEMPRET.M with the
given input arguments.
%
%   GRAFICASFUERZTIEMPRET('Property','Value',...) creates a new
GRAFICASFUERZTIEMPRET or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before GraficasFuerzTiempRet_OpeningFcn gets
called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to
GraficasFuerzTiempRet_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
```

```
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
GraficasFuerzTiempRet

% Last Modified by GUIDE v2.5 20-Nov-2011 11:56:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @GraficasFuerzTiempRet_OpeningFcn, ...
                  'gui_OutputFcn',    @GraficasFuerzTiempRet_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GraficasFuerzTiempRet is made visible.
function GraficasFuerzTiempRet_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to GraficasFuerzTiempRet (see
VARARGIN)

% Choose default command line output for GraficasFuerzTiempRet
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GraficasFuerzTiempRet wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GraficasFuerzTiempRet_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
function nodofu_Callback(hObject, eventdata, handles)
% hObject      handle to nodofu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodofu as text
%         str2double(get(hObject,'String')) returns contents of nodofu
as a double

% --- Executes during object creation, after setting all properties.
function nodofu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodofu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function colfu_Callback(hObject, eventdata, handles)
% hObject      handle to colfu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of colfu as text
%         str2double(get(hObject,'String')) returns contents of colfu
as a double

% --- Executes during object creation, after setting all properties.
function colfu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to colfu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ejefu_Callback(hObject, eventdata, handles)
% hObject      handle to ejefu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ejefu as text
%         str2double(get(hObject,'String')) returns contents of ejefu
as a double
```

```
% --- Executes during object creation, after setting all properties.
function ejefu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ejefu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject      handle to aceptar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global numero_nodos t ans

contadorfuerz=0;
for i=1:numero_nodos
    for I=1:3
        contadorfuerz=contadorfuerz+1;
        matfuerz(i,I)=contadorfuerz;
    end
end

%El numero de nodo elegido por el usuario se asigna a la variable
nodofuerz
nodofuerz=str2double(get(handles.nodofu,'String'));
%El numero de eje elegido por el usuario se asigna a la variable
ejefuerz
ejefuerz=str2double(get(handles.ejefu,'String'));
%%El numero de color elegido por el usuario se asigna a la variable
%colorfuerz
colorfuerz=str2double(get(handles.colfu,'String'));
%Bucle para que las graficas salgan de un determinado color

if colorfuerz==1
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'g');
elseif colorfuerz==2
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'r');
elseif colorfuerz==3
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'c');
elseif colorfuerz==4
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'m');
elseif colorfuerz==5
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'y');
elseif colorfuerz==6
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'k');
elseif colorfuerz==7
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'--');
else
plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:));
end

%Se da titulo a la grafica y a los x e y
title('Fuerza-Tiempo');
```

```
xlabel('Tiempo');
ylabel('Fuerza');
guidata(hObject, handles);

% --- Executes on button press in limpiar.
function limpiar_Callback(hObject, eventdata, handles)
% hObject    handle to limpiar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se limpia el axes de las graficas que esten dibujadas
cla(handles.axes1,'reset')
guidata(hObject, handles);

% --- Executes on button press in holdoff.
function holdoff_Callback(hObject, eventdata, handles)
% hObject    handle to holdoff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Las graficas se dejan de superponer en el axes

hold off

% --- Executes on button press in holdon.
function holdon_Callback(hObject, eventdata, handles)
% hObject    handle to holdon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se mantienen las graficas en el axes, haciendo que se puedan
superponer
%unas graficas con otras
hold on

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
% hObject    handle to close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se cierra la ventana GraficasFuerzTiempRet
close GraficasFuerzTiempRet

% --- Executes on button press in limpiarfuerz.
function limpiarfuerz_Callback(hObject, eventdata, handles)
% hObject    handle to limpiarfuerz (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se eliminan todos lo datos introducidos y calculados

clear all
```

## Resultados en gráficas de fuerzas dinámicas en estructuras articuladas

```
function varargout = GraficasFuerzTiempRet(varargin)
% GRAFICASFUERZTIEMPRET M-file for GraficasFuerzTiempRet.fig
%   GRAFICASFUERZTIEMPRET, by itself, creates a new
%   GRAFICASFUERZTIEMPRET or raises the existing
%   singleton*.
%
%   H = GRAFICASFUERZTIEMPRET returns the handle to a new
%   GRAFICASFUERZTIEMPRET or the handle to
%   the existing singleton*.
%
%   GRAFICASFUERZTIEMPRET('CALLBACK',hObject,eventData,handles,...)
```

```
calls the local
%      function named CALLBACK in GRAFICASFUERZTIEMPRET.M with the
given input arguments.
%
%      GRAFICASFUERZTIEMPRET('Property','Value',...) creates a new
GRAFICASFUERZTIEMPRET or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before GraficasFuerzTiempRet_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
GraficasFuerzTiempRet_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
GraficasFuerzTiempRet

% Last Modified by GUIDE v2.5 20-Nov-2011 11:56:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @GraficasFuerzTiempRet_OpeningFcn, ...
                  'gui_OutputFcn',  @GraficasFuerzTiempRet_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GraficasFuerzTiempRet is made visible.
function GraficasFuerzTiempRet_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to GraficasFuerzTiempRet (see
VARARGIN)

% Choose default command line output for GraficasFuerzTiempRet
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```



```
% UIWAIT makes GraficasFuerzTiempRet wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GraficasFuerzTiempRet_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function nodofu_Callback(hObject, eventdata, handles)
% hObject      handle to nodofu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nodofu as text
%          str2double(get(hObject,'String')) returns contents of nodofu
as a double

% --- Executes during object creation, after setting all properties.
function nodofu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to nodofu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function colfu_Callback(hObject, eventdata, handles)
% hObject      handle to colfu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of colfu as text
%          str2double(get(hObject,'String')) returns contents of colfu
as a double

% --- Executes during object creation, after setting all properties.
function colfu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to colfu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ejefu_Callback(hObject, eventdata, handles)
% hObject    handle to ejefu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ejefu as text
%         str2double(get(hObject,'String')) returns contents of ejefu
as a double

% --- Executes during object creation, after setting all properties.
function ejefu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ejefu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in aceptar.
function aceptar_Callback(hObject, eventdata, handles)
% hObject    handle to aceptar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global numero_nodos t ans

contadorfuerz=0;
for i=1:numero_nodos
    for I=1:3
        contadorfuerz=contadorfuerz+1;
        matfuerz(i,I)=contadorfuerz;
    end
end

nodofuerz=str2double(get(handles.nodofu,'String'));
ejefuerz=str2double(get(handles.ejefu,'String'));
colorfuerz=str2double(get(handles.colfu,'String'));
if colorfuerz==1
    plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'g');
elseif colorfuerz==2
    plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'r');
elseif colorfuerz==3
    plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'c');
elseif colorfuerz==4
    plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'m');
elseif colorfuerz==5
    plot(t,ans(matfuerz(nodofuerz,ejefuerz,:),:),'y');
elseif colorfuerz==6
```

```
plot(t,ans(matfuerz(nodofuerz,ejefuerz),:),'k');
elseif colorfuerz==7
plot(t,ans(matfuerz(nodofuerz,ejefuerz),:),'--');
else
plot(t,ans(matfuerz(nodofuerz,ejefuerz),:));
end

title('Fuerza-Tiempo');
xlabel('Tiempo');
ylabel('Fuerza');
guidata(hObject, handles);

% --- Executes on button press in limpiar.
function limpiar_Callback(hObject, eventdata, handles)
% hObject    handle to limpiar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1,'reset')
guidata(hObject, handles);

% --- Executes on button press in holdoff.
function holdoff_Callback(hObject, eventdata, handles)
% hObject    handle to holdoff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
hold off

% --- Executes on button press in holdon.
function holdon_Callback(hObject, eventdata, handles)
% hObject    handle to holdon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
hold on

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
% hObject    handle to close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
hold on
close GraficasFuerzTiempRet

% --- Executes on button press in limpiarfuerz.
function limpiarfuerz_Callback(hObject, eventdata, handles)
% hObject    handle to limpiarfuerz (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clear all
```

## Resultados numéricos de fuerzas en dinámico en estructuras reticuladas

```
function varargout = ReaccionesER_Dinamico(varargin)
% REACCIONESER_DINAMICO M-file for ReaccionesER_Dinamico.fig
% REACCIONESER_DINAMICO, by itself, creates a new
REACCIONESER_DINAMICO or raises the existing
% singleton*.
%
% H = REACCIONESER_DINAMICO returns the handle to a new
REACCIONESER_DINAMICO or the handle to
% the existing singleton*.
%
% REACCIONESER_DINAMICO('CALLBACK',hObject,eventData,handles,...)
```

```
calls the local
%      function named CALLBACK in REACCIONESER_DINAMICO.M with the
given input arguments.
%
%      REACCIONESER_DINAMICO('Property','Value',...) creates a new
REACCIONESER_DINAMICO or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before ReaccionesER_Dinamico_OpeningFcn gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
ReaccionesER_Dinamico_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
ReaccionesER_Dinamico

% Last Modified by GUIDE v2.5 19-Nov-2011 21:23:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @ReaccionesER_Dinamico_OpeningFcn, ...
                  'gui_OutputFcn',    @ReaccionesER_Dinamico_OutputFcn,
                  ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ReaccionesER_Dinamico is made visible.
function ReaccionesER_Dinamico_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to ReaccionesER_Dinamico (see
VARARGIN)

% Choose default command line output for ReaccionesER_Dinamico
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes ReaccionesER_Dinamico wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ReaccionesER_Dinamico_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global ReaccionesDinamicoReticulada t numero_nodos
%Obtencion de un vector cuyos valores serviran para entender mejor los
%resultados obtenidos, pues que indicaran el nodo y el sentido de la
fuerza
contadorreacc=0;
for i=1:numero_nodos
    for I=1:3

        contadorreacc=contadorreacc+1;
        matrizindicativaRe(contadorreacc,:)=(10*i)+I;
    end
end

matrizindicativaRe=[0;matrizindicativaRe];

resultadosreaccionesdinamicoreticulada=[t';ReaccionesDinamicoReticulad
a];

%Se muestran los resultados de las reacciones en la tabla

set(handles.uitable1,'data',resultadosreaccionesdinamicoreticulada)
%Se nombra a las filas de la tabla segun el vector matrizindicativaAr

set(handles.uitable1,'RowName',matrizindicativaRe)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
% hObject      handle to cerrar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Se cierra la ventana ReaccionesER_Dinamico
close ReaccionesER_Dinamico
```

### Resultados numéricos de fuerzas en dinámico en estructuras articuladas

```
function varargout =
Resultados_Reacciones_Dinamico_Articuladas(varargin)
% RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS M-file for
Resultados_Reacciones_Dinamico_Articuladas.fig
%     RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS, by itself, creates
a new RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS or raises the
existing
%     singleton*.
%
```

```
%      H = RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS returns the
handle to a new RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS or the
handle to
%      the existing singleton*.
%
%
RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS('CALLBACK',hObject,eventDat
a,handles,...) calls the local
%      function named CALLBACK in
RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS.M with the given input
arguments.
%
%
RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS('Property','Value',...)
creates a new RESULTADOS_REACCIONES_DINAMICO_ARTICULADAS or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before
Resultados_Reacciones_Dinamico_Articuladas_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
Resultados_Reacciones_Dinamico_Articuladas_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Resultados_Reacciones_Dinamico_Articuladas

% Last Modified by GUIDE v2.5 19-Nov-2011 21:21:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Resultados_Reacciones_Dinamico_Articuladas_OpeningFcn, ...
                  'gui_OutputFcn',    @Resultados_Reacciones_Dinamico_Articuladas_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Resultados_Reacciones_Dinamico_Articuladas
is made visible.
function
Resultados_Reacciones_Dinamico_Articuladas_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to
Resultados_Reacciones_Dinamico_Articuladas (see VARARGIN)

% Choose default command line output for
Resultados_Reacciones_Dinamico_Articuladas
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Resultados_Reacciones_Dinamico_Articuladas wait for
user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout =
Resultados_Reacciones_Dinamico_Articuladas_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

global t ReaccionesDinamicoArticuladas numero_nodos

%Obtencion de un vector cuyos valores serviran para entender mejor los
%resultados obtenidos, pues que indicaran el nodo y el sentido de la
fuerza
contadorreacc=0;
for i=1:numero_nodos
    for I=1:2

        contadorreacc=contadorreacc+1;
        matrizindicativaAr(contadorreacc,:)=(10*i)+I;
    end
end

matrizindicativaAr=[0;matrizindicativaAr];

reaccionestiempodinamicoarticualdas=[t';ReaccionesDinamicoArticuladas]
;

%hay que meter un bucle para identificar que apoyo y que
desplazamiento
%queremos indicar

%Se muestran los resultados de las reacciones en la tabla
set(handles.reaccionesdinamicoarticualdas_output,'data',reaccionestiem
podinamicoarticualdas)
%Se nombra a las filas de la tabla segun el vector matrizindicativaAr
set(handles.reaccionesdinamicoarticualdas_output,'RowName',matrizindic
ativaAr)

% --- Executes on button press in cerrar.
function cerrar_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to cerrar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%Se cierra la ventana close Resultados_Reacciones_Dinamico_Articuladas

close Resultados_Reacciones_Dinamico_Articuladas
```